Management of the Internet and Complex Services

European Sixth Framework Network of Excellence FP6-2004-IST-026854-NoE

Deliverable 8.2
Development of a Multi-provider Model, an SLM Model, and Service Provisioning Concepts

The EMANICS Consortium

Caisse des Dépôts et Consignations, CDC, France
Institut National de Recherche en Informatique et Automatique, INRIA, France
University of Twente, UT, The Netherlands
Imperial College, IC, UK
International University Bremen, IUB, Germany
KTH Royal Institute of Technology, KTH, Sweden
Oslo University College, HIO, Norway
Universidat Politecnica de Catalunya, UPC, Spain
University of Federal Armed Forces Munich, CETIM/UniBwM, Germany
Poznan Supercomputing and Networking Center, PSNC, Poland
University of Zürich, UniZH, Switzerland
Ludwig-Maximilian University Munich, LMU, Germany
University of Surrey, UniS, UK
University of Pitesti, UPI, Romania

For more information on this document or the EMANICS project, please contact:

Dr. Olivier Festor
Technopole de Nancy-Brabois — Campus scientifique
615, rue de Jardin Botanique — B.P. 101
F—54600 Villers Les Nancy Cedex
France

Phone: +33 383 59 30 66
Fax: +33 383 41 30 79
E-mail: <olivier.festor@loria.fr>

Document Control

Title: Development of a Multi-provider Model, an SLM Model, and Service Provisioning Concepts

Type: Public

Editors: Burkhard Stiller, David Hausheer

E-mail: stiller@ifi.uzh.ch, hausheer@ifi.uzh.ch

Authors: Mark Burgess, Stylianos Georgoulas, David Hausheer, Iris Hochstatter, Thomas Schaaf, Gregor Schaffrath, Burkhard Stiller, Martin Waldburger (alphabetic order)

Doc ID: D8.2-v1.2

AMENDMENT HISTORY

Version	Date	Author	Description/Comments
V0.1	March 13, 2007	Martin Waldburger (MW)	Document structure, External Effects of Multi-provider Scenarios
V0.2	March 13, 2007	David Hausheer (DH)	Update of content structure
V0.3	March 14, 2007	Burkhard Stiller (BS)	Extension of ToC to pre-final state, overviews, partly introductions written
V0.4	March 16, 2007	Stylianos Georgoulas (SG), Gregor Schaffrath (GS), Thomas Schaaf (TS), BS	Section 4.2 input, Section 3 and 4.1 input, Section 5 structure update, editorial updates
V0.5	April 3, 2007	Mark Burgess (MB), Thomas Schaaf, Iris Hochstatter (IH), MW	Section 5 input, Section 6 input, Section 5 structure update, editorial updates
V0.6	April 16, 2007	TS, GS, BS	Section 5 input, Section 4.2 input, spell-check, editorial updates
V0.7	April 18, 2007	SG, BS	Section 6 input, editorial updates
V0.8	April 22, 2007	GS, IH, BS	Section 3 Update, Section 4.1 completed, Section 6 included, executive summary
V0.9	June 13, 2007	Jan Gerke, MW, GS, IH, TS, SG, MB, BS	Updates of basically all sections and inclusion of input from all partners
V1.0	June 16, 2007	SG, IH, DH, BS	Finalization of almost all missing pieces of text and paragraphs, many editorial activities, update of references, spell-check, summary and conclusions
V1.1	June 29, 2007	BS, SG, MB, IH, TS	Corrections, input, and completion of all sections
V1.2	July 3, 2007	BS, DH, MW, DH, GS, TS, MB	Last corrections and updates included, submission ready state reached

Table of Content

1 Executive Summary

Managing the Internet as well as its related services and traffic determines the important goal, thus, EMANICS in general and WP8 on Economic Management specifically intends to understand, to determine interrelations, and to simplify such Internet management tasks. This area of concern has been addressed here explicitly under an integrated point of view of multi-domain models, Service Level Management (SLM), and service provisioning.

While the provisioning of network services in the Internet requires, besides suitable technology and mechanisms, a well-defined set of economic measures and means in order to be economically viable, its multi-domain aspects of collaborating and competing providers on the transport and services level forms the major challenge of today's Internet. Thus, to be able to achieve a scalable solution, only an approach addressing technological and economic perspectives of a management framework may succeed in the future. The main reason for this statement originates from the competition of network service providers, third party providers, covering application and content providers, and multiple niche service providers with a large variety of technologies and business models in operation or under preparation. The SLM approach delivers in this context monitoring and reporting tasks, which follow an optimization of economic and technical goals of an entire organization and an improved relationship between providers and customers. To address specify, support, and enable these technical and economic dimensions of Internet management in an integrated manner, the basis is required for providing an appropriate benefit for these providers and such an approach meets the needs of many customers at the same time.

The work addresses multi-provider dimensions of network and service management in a structured manner. Based on multi-provider dimensions—covering technical, economic, and operational aspects—external effects of multi-provider scenarios are investigated. This reveals that externalities are of high importance for Internet services, parties involved, and property rights under question. Since the absence of transaction costs in a multi-provider case typically does not hold true, the higher number of providers involved does increase bargaining costs, which in turn leads to a potential externality compensation requirement. Thus, a correct assignment of resource property right decreases.

Additionally, the work done here covers SLM investigations in the light of customer-oriented service management and provisioning of high-quality Information Technology (IT) services. In this context modeling and formalization of Service Level Agreements (SLA) determines the basis for a multi-provider/multi-customer relationship, which addresses negotiations, content, and business alignments. An approach on promise theory applied to SLA management shows a path to address voluntary cooperation under stated assumptions.

Finally, the area of provisioning concepts for Internet services addresses Quality-of-Service (QoS) provisioning, respective policies, and their application in practice. Useful provisioning patterns have been observed, however, many provider-specific and service-dependent schemes are applied. While overprovisioning still seems to dominate the provider environment, the balance between effort and costs starts to be investigated based on IT Infrastructure Library (ITIL) guidelines and best practices.

Therefore, EMANICS' deliverable D8.2 outlines in a detailed manner steps and actions undertaken to integrate economic and technical management mechanisms in a homogeneously interacting approach. Additionally, a set of joint papers attached cover some of those important details of the work sketched in D8.2. Still, future steps are required to complete a fully concise approach. In particular, a critical assessment and evaluation of the identified models, concepts, and mechanisms is foreseen for EMANICS phase two.

(This page was left empty intentionally.)

2 Introduction

Providers and customers will exist in a multi-domain environment with multiple boundaries to be crossed, such as administrative, legal, networking, and topology boundaries. Thus, the focus on such overlapping points by respective mechanisms will be successful, if Service Level Management (SLM) principles are extended and service provisioning steps are supported. Driven by the goals to be achieved by a service provisioning approach under economic measures as well as technical Service Level Agreement (SLA) mechanisms, such boundaries can be tackled and adequate approaches are to be developed.

Therefore, this deliverable D8.2 shows a number of key interrelations and dependencies with deliverable D8.1, “Definition of Service Provisioning Goals, Economic Impacts, and SLA Management Tasks”. The work on D8.1 was mainly motivated by identifying scalable technology and mechanisms in support of economic service deployment and provisioning in tomorrow’s Internet. In that context, the management of heterogeneous and partly broadband-capable infrastructure is a key task.

Based on this extensive state-of-the-art study in the respective areas of traditional network management and basic economics, newly developed integrated concepts for network management were presented in a certain level of detail. This lead to an in-depth discussion of the key set of emerging research questions for tomorrow’s network management and technology requirements in full support of future economic management schemes.

These detailed considerations revealed the urgent need to develop advanced network management techniques. Such techniques are required to meet the following requirements:

To be based on well-defined roles and models.

To cover the full set of promises and interactions.

To enable multiple parties—distributed across domains—in accessing infrastructure in an economically viable and fair way.

Driven by these requirements, 5 central, inter-dependent models for network management were defined and specified in D8.1. In a role model, players are visualized and analyzed with respect to their respective behavior in a multi-provide/multi-user context. A service model determines the types of services envisioned, while a charging model describes those activities required to pay for service usage a fair and price announced. Business models, shaped by those three previously named models, find instantiations for operations in deployment models.

Those main models formed the basis for the definition and discussion of several key mechanisms in network management. In particular, SLA as a means to express service quality expectations and the respective delivery actions were discussed in detail. This lead to a comprehensive investigation on the service provisioning processes and schemes for service negotiation between service providers and service users, possibly guided by policies. This was complemented by presenting mechanisms for technical accounting, constituting the basis for charge calculation determined by accounted resource usage records, defined accounting actions and events.

Here, Deliverable D8.2 takes those results and preliminary conclusions drawn as input to achieve its mission, which is defined by the development of a multi-provider model, a SLM model, and service provisioning concepts. Building partly on D8.1 and elaborating its results into an integrated concept for network management, it leverages key service provisioning concepts, such as Quality-of-Service (QoS), in a multi-domain environment, based on the full set of required and specified SLM techniques.

2.1 Purpose of the Document D8.2

Deliverable D8.2 of Work Package 8 within EMANICS continues to describe the economic dimensions of network management in detail, with respect to technology and mechanisms required for an operationally effective and economically efficient future network management approach. Thus, D8.2 addresses three key questions for the reader.

First, the multi-provider problem in real networks affects the network management tasks in terms of technology and mechanisms and in terms of economic control as discussed in D8.1. Therefore, the clear multi-domain aspects are analyzed and their influences on network management are presented. Second, the SLM required to fulfill service guarantees in a given environment, specifically addressing the end-to-end and multi-domain situation, is introduced. A relevant framework and formalization for the application of these components are given. Third, the provisioning of services in a multi-domain environment becomes the important task of providers to offer services in a commercial manner. Therefore, the relevant provisioning mechanisms, including QoS handling and its negotiations, are described. Those are complemented with respective policies to address the flexibility in the multi-domain case.

Thus, the readership envisioned for D8.2 covers EMANICS partners providing them with a clear view on the multi-provider problem of interacting providers. which form the basis for many steps undertaken in EMANICS. On the other hand, the document is addressed toward the network management community as a whole, documenting advantages in a full and transparent integration of economic management techniques into traditional models of network management.

2.2 Document Outline

The starting point for work in the context of the multi-provider scenarios in real networks, the Service Level Management (SLM) approach, and the service provisioning concept in this respect is determined in Section 3, where the core set of related work is briefly summarized.

Secondly, Section 4 introduces the multi-domain aspects and discusses their technological and economic situation.

Thirdly, the SLM approach is presented in Section 5 in terms of a framework and its formalization.

Fourthly, Section 6 outlines key issues of service provisioning in a multi-domain environment, while addressing frameworks, policies, and provisioning in practice.

Finally, Section 7 wraps up and draws preliminary conclusions.

D8.2 is complemented in Section 8 with a glossary, in Section 9 with major bibliographic references, in Section 11 with acknowledgements, and in Section 12 with a selection of cooperative work undertaken by EMANICS' WP8 partners.

3 Related Work

Related work on D8.2 addresses in brief architectures in support of reaching network QoS, multi-domain issues for IP-based communications, Service Level Management models, and service provisioning approaches. Each of those selected approaches is introduced and key features of importance in the context of D8.2 is discussed.

3.1 Architectures for Network QoS

The overprovisioning model requires that the ratio of traffic demand to the available resources remains small [49]. Providers set small thresholds of link utilizations and when these thresholds are crossed, the links are considered congested and, therefore, their capacity is upgraded. There is anecdotal evidence that these utilization thresholds are as low as 15%. The excess capacity in the network absorbs the transient bursts of traffic, allowing for QoS demanding services to be offered to end-users.

In order to support QoS in the Internet in a more predictable and also resource and, consequently, cost-efficient manner, the Internet Engineering Task Force (IETF) has defined two architectures: the Integrated Services (IntServ) [40] and the Differentiated Services (DiffServ) [5] architecture.

3.1.1 The Integrated Services Architecture (IntServ)

The Integrated Services architecture follows an approach similar to that found in multiservice telecommunication networks, most notably in Asynchronous Transfer Mode (ATM) networks. In this architecture there is a hard sense of QoS in terms of resources allocated to individual flows, with the Resource Reservation Protocol (RSVP) used for signalling the required QoS characteristics to the network.

With flow state information required at every router in the path between receiver and transmitter, scalability has been the main architectural concern and one of the main reasons that restricted its deployment. The amount of state information increases proportionally with the number of flows, thus placing a huge storage and processing overhead to the routers and requiring fairly complex control components in each router.

The Integrated Services architecture supports two new classes of service in addition to the existing best effort class. These are the Guaranteed and the Controlled Load service classes. The Guaranteed Service class is a quantitative service, which provides strong guarantees in terms of end-to-end delay and bandwidth. It also ensures that no packets will be discarded due to queues overflowing anywhere in the network. The Controlled Load service is a qualitative service and it is defined as being equivalent to the service obtained using best effort on a lightly loaded network. If the load on the network increases the best effort traffic will find its service quality degraded while Controlled Load traffic will still receive the service it got under the light load scenario.

3.1.2 The Differentiated Services Architecture (DiffServ)

The Differentiated Services architecture, as proposed by the IETF (Internet Engineering Task Force) Differentiated Services Working Group, allows IP traffic to be classified into a finite number of service classes that receive different router treatment. For example, traffic belonging to a higher priority and/or delay service class receives some form of preferential treatment over traffic classified into a lower service class.

Differentiated services do not attempt to give explicit end-to-end guarantees. Instead, in congested network elements, traffic with a higher class of priority has a higher probability of getting through, or in case of delay priority, is scheduled for transmission before traffic that is less delay-sensitive. DiffServ (DS) follows a “keep all complexity at the network edges” approach where all complicated per-flow packet processing is done at the network boundaries. The information required to perform actual differentiation in the network elements is carried in the Type of Service (ToS) field of the IPv4 packet headers or the Traffic Class field of the IPv6 packet headers, referred to as the DS Field or Code Point

(DSCP). Thus, since the information required by the buffer management and scheduling mechanisms is carried within the packet, the amount of state information, which is required to be maintained per node, is proportional to the number of service classes and not proportional to the number of flows.

The Differentiated Services architecture supports two new classes of service in addition to the existing best effort class. These are the Expedited Forwarding (EF) service class and the Assured Forwarding (AF) service class. The Expedited Forwarding service class emulates a “virtual wire” environment, whereas the Assured Forwarding service class provides more “softer” guarantees than EF and can be further divided to four sub-classes, AF1-AF4 with three-drop precedence levels within each of them.

3.2 Multi-domain Work

Due to the distributed nature of today’s Internet and the large number of services that can benefit from its widespread infrastructure and run on top of it, there need to exist numerous relationships between a multitude of stakeholders who are responsible, each one for part of the provision of end-to-end connectivity and value-added services. These relationships need to be well defined in order for the Internet to be able to provide QoS enabled services and also need to span the whole end-to-end QoS chain.

Towards this direction, many projects have worked and still are currently working in this area, aiming to specify the stakeholders and the relationships between them that are needed for offering end-to-end QoS enabled services. Examples of projects falling in this area were/are the MESCAL, AGAVE, and ENTHRONE projects, which are outlined below.

3.2.1 MESCAL

MESCAL (Management of End-to-end Quality of Service Across the Internet at Large) [31] was one of the first projects working in the direction of end-to-end QoS provisioning. MESCAL addressed two major issues for inter-domain QoS delivery. The first one was the definition of QoS-based connectivity services to be provided by stakeholders. To this end, the stakeholders and the relationships between them were identified by extending the notion of SLA (Service Level agreement) to account for relationships between customers and providers (cSLA) and between providers themselves (pSLA). The second issue was the means to engineer the resources to meet agreed performance and capacity targets for the contracted services. To this end, the project proposed algorithms for offline and online/ dynamic inter-domain traffic engineering and provisioning. A MESCAL-compliant service provisioning template showing the stakeholders and relationships in the end-to-end QoS chain as identified in the context of the MESCAL project can be found in Section 4.2.3 of Emanics Deliverable 8.1.

3.2.2 AGAVE

AGAVE (A Lightweight Approach for Viable End-to-end IP-based QoS Services) [1] is a follow-up of the MESCAL project and is an ongoing project working in the area of multi-domain QoS delivery. Contrary to MESCAL, where there was no distinction among the roles of Application-level Service Providers (ASPs) and IP Connectivity Providers (ICPs), AGAVE is explicitly distinguishing the roles of (ASPs) and ICPs, separating the service and network concerns. It is working on specifying the relationships and interactions between the stakeholders in such an environment, where ASPs and ICPs can be completely disjoint entities, to enable end-to-end connectivity and service provisioning. An AGAVE-compliant

service provisioning template showing the stakeholders and relationships in the end-to-end QoS chain as identified in the context of the AGAVE project will be provided in Section 4.3 of this Deliverable.

3.2.3 ENTHRONE

ENTHRONE (End-to-End QoS through Integrated Management of Content, Networks and Terminals) [17], currently at its second phase, is a project working in the definition of stakeholders and their relationships in order to enable QoS-enabled delivery of multimedia content that needs to span several domains from the content server till the content consumers. ENTHRONE, therefore, does not attempt to provide a business/service model generic enough to cover many types of service; instead it attempts to provide a complete solution for the end-to-end provisioning of this specific type of service. An ENTHRONE-compliant service provisioning template showing the stakeholders and their interactions as identified by ENTHRONE project in the end-to-end content delivery chain will be provided in Section 4.3 of this Deliverable.

3.2.4 Multi-provider Roaming Support

The work performed in DAMMO and DAMMO II (Distributed Accounting and Auditing Management for Multiple Mobile Network Operators) [14] aims at facilitating the accounting in multi-domain scenarios, by describing a distributed framework for A4C (Authentication, Authorization, Accounting, Auditing, and Charging) configuration and communication in a multi-domain network service scenario, considering mobility aspects.

It assumes low level mechanisms to be in place and focuses on the elaboration of

Diameter protocol extensions, supporting the transport of accounting related configuration and record information across domain boundaries,

A distribution framework of components in multi-domain environments, enabling arbitrary cooperation between multiple providers,

A communication protocol to handle accounting tasks both in the service setup and teardown phase and inter-/intra-handover scenarios, and

A service and session model in support of accounting related operations.

Flexibility considerations and requirement derivations are based on a generic role model as well as investigations into different business models.

A prototypical implementation has been done, which is based on the scenario in Figure 1:

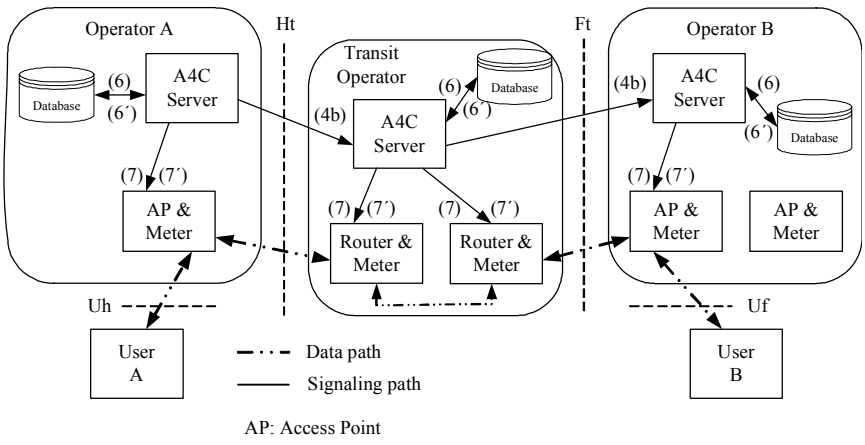


Figure 1: DAMMO Multi-provider Scenario (based on [14])

3.3 SLM Work

Service Level Management is considered a vital discipline for successful IT and IT Service Management (ITSM) by most IT and telecommunication providers in terms of enterprises or data centers. This is basically due to the fact that effective Quality of Service (QoS) management provides the necessary foundations in today's common situation of a steadily increasing focus on business alignment of IT operations, where IT departments are emerging from cost centers to profit centers.

To face the various challenges of Service Level Management, different approaches have evolved in the last decade. Some of them come with concepts of great relevance for current and future research efforts in the discipline of Economic Management. In the following, these approaches are briefly introduced and described as the current state-of-the-art in Service Level Management. The differentiation is done between Management Frameworks, Service Composition and Negotiation, Service Negotiation Protocols, Performance Management and the Management by Contract (MbC) paradigm.

3.3.1 Management Frameworks

In general, Management Frameworks provide concepts and tools in support of the execution of management tasks. Frameworks introduced in the following are not limited to SLM-specific issues, but address this topic as one amongst other management disciplines.

IT Infrastructure Library (ITIL)

ITIL (IT Infrastructure Library) is a collection of books, in which best practices in ITSM are described. Today, ITIL can be seen as a de-facto standard in the discipline of ITSM, for which it provides guidelines by its current core titles Service Support [34] and Service Delivery [35].

ITIL follows the principle of process-oriented (IT Service-) Management: Every management activity taking place within an IT organization is part of one of the defined management processes. Thus, process-orientation extends the idea of functional management where IT management decisions and actions take place in different departments (*e.g.*, network department, server department, storage department). In effect, the responsibilities for specific IT management decisions can be shared between different organizational units as the management processes span the entire IT organization independent from its organizational partition.

Service Level Management is one of ITIL's core processes. Its goal is described by ITIL as to improve the overall service quality and represent the IT organization towards its customers. An overview and analysis of this process and the related measures and activities is given in Section 5.1.2.1.

Control Objectives for Information and related Technology (COBIT)

COBIT is an important framework for IT governance and claims for itself to bridge the gap between control objectives, IT Service Management, technical issues and business risks [27]. It has been released by the Information Systems Audit and Control Association (ISACA) and is continually revised.

COBIT is (like ITIL) a process-oriented framework and arranges its IT activities using four domains: Plan and Organize, Acquire and Implement, Deliver and Support (DS) and Monitor and Evaluate. Defining and managing Service Levels is one activity of the DS domain (DS1). Described control objectives are similar to the concepts described by ITIL.

New Generation Operations Systems and Software (NGOSS)

NGOSS is a programme of the TeleManagement Forum (TMF) which aims at providing guidance especially for Telecommunication Service Providers to manage their entire business [44]. It is not, as ITIL and COBIT, limited to the management activities in the field of IT operations. One of the various publications coming from the NGOSS project is the SLA Management Handbook of which an overview is given in Section 5.1.2.2.

3.3.2 Service Composition and Negotiation

Service composition is the process of creating new value-added services out of already existing services. Service composition has numerous advantages, such as supporting rapid deployment, effective resource usage and creating new business opportunities. As a step towards fully automated service composition, its underlying mechanisms have been researched and improved.

First, in order for a composed service to be reliable, its component services must be provided in a reliable manner. To this end, legally enforceable Service Level Agreements (SLAs) are required. A service negotiation mechanism for use between a service provider and a service consumer has been designed, which results in such an SLA [23]. The negotiation mechanism makes use of an encrypted communication channel and ensures the non-repudiation of service offers and the final SLA through signatures and a trusted third party where complaints can be files.

Second, in order to carry out service composition in an automated manner, an algorithm for composing services is required. Such an algorithm as well as different heuristic strategies for applying it have been designed and have been evaluated through simulations, in which a distributed computing service was composed out of smaller computing services [22]. As a result, guidelines for creating new heuristic strategies have been derived.

Third, at least in the near future, due to its complexity, service composition will be carried out by a small number of specialized service composers. Thus, they will form an oligopoly, without social welfare of service composition being automatically maximized through perfect competition. In order to maximize the social welfare of service competition, a new pricing mechanism has been designed in the form of VETO, the Vickrey welfare procurement auction [24]. VETO ensures that each service composer's optimal strategy is to bid his true valuation of a composed service's welfare. Furthermore, it allocates fair shares of welfare created by the winning bid to both, service composer and service consumer. To do this, VETO only requires one single round of bidding. The outcome of the auction is verifiable by all its participants and it can be detected and proven if a participant attempts to cheat. Finally, VETO leads to an enforceable contract, without revealing details of a bid to competitors.

3.3.3 Service Negotiation Protocols

The negotiation of services and respective parameter sets determines an important basis for dealing with and handling Service Level Agreements. Thus, the following approaches are summarized at this stage.

Resource Negotiation and Pricing Protocol (RNAP)

RNAP [51] was one of the earliest protocols developed to facilitate dynamic service negotiation and can be considered an extension of Resource Reservation Protocol (RSVP). RNAP employs a soft state approach to service negotiation. A distinguishing characteristic of RNAP is that it allows the negotiation of price of the contracted services.

Service Negotiation Protocol (SrNP)

A unique feature of SrNP [48] is that the protocol is not specific to any particular SLS format or to the context of an SLS. It is general enough to be applied for negotiating any document provided that the document is in the form of attribute-value pairs. SrNP can be implemented by extending either the RSVP or COPS protocols.

Common Open Policy Service Protocol—Service Level Specification (COPS-SLS)

COPS-SLS [32] is an extension of the COPS protocol to negotiate an SLS. A characteristic feature of it is that it distinguishes the interactions between a customer and an ASP into two phases: configuration and negotiation. In the configuration phase the ASP informs the customer how to request a level of service and in the negotiation phase the customer uses the information from the configuration phase to request the service.

Dynamic Service Negotiation Protocol (DSNP)

Unlike RNAP, SrNP and COPS-SLS that are extensions of other existing protocols, DSNP [10] was developed exclusively for dynamic service negotiation. Consequently, it is lightweight and better suited, therefore, to wireless devices, such as PDAs and mobile phones.

QoS NSIS Signaling Layer Protocol (QoS-NSLP)

QoS-NSLP [52] is a protocol for signaling QoS reservations in the Internet. It is defined by IETF's NSIS working group as an extension of RSVP. Its operation is quite similar to RNAP and it too employs a soft state approach to service negotiation.

QoS Generic Signaling Layer Protocol (QoS-GSLP)

QoS-GSLP [2] is proposed by the Ambient Networks Consortium and is used for controlling and negotiating bilateral QoS requirements in wireless environments. It builds on top of IETF's NSIS protocol suite and uses knowledge about mobility and traffic patterns to setup SLSes well in advance.

3.3.4 Performance Measurements: Indicators and SLA Monitoring

Performance measurements are needed in any organization to measure the progress towards the goals/targets set when introducing a service. The performance metrics that need to be defined are each time service-dependent. Moreover SLA monitoring needs to be in place to verify that the level of offered service meets the pre-agreed level of service, as appears in SLAs. The details of SLA monitoring are also service-dependent.

With respect to the performance metrics, the notion of Key Performance Indicators is a term generic enough that can be defined on a per-service basis. With respect to SLA monitoring, IBMs WSLA framework describes a generic enough process showing the logic and targets of an SLA monitoring system; logic and targets which can be valid for a variety of services.

Key Performance Indicators (KPIs)

Key Performance Indicators (KPI), also known as Key Success Indicators (KSI), help an organization define and measure progress towards its goals [38]. They are quantifiable measurements, agreed beforehand, that reflect the critical success factors of an organization and they differ depending on the organization.

Whatever KPI are selected, they must reflect the organization's goals, they must be key to its success and they must be quantifiable (measurable). KPIs usually reflect long-term

considerations. The definition of what they are and how they are measured do not change often. The goals of a particular KPI may change as the organization's goals change, or as it gets closer to achieving a goal.

IBM's WSLA Framework

IBM's WSLA (Web Service Level Agreement) framework [28], with respect to SLA monitoring, consists of three services. The Measurement service probes and measures resource metrics according to the SLA specification and aggregates them into SLA parameters. The Condition Evaluation Service compares the SLA parameters obtained by the Measurement Service against specified service levels and produces notifications in case a violation has occurred during a time period (which is considered to be valid for these types of measurements). The third service, the Management Service, carries out corrective actions, provided they do not violate business policies. That means that feasible corrective actions may not be carried out if they contradict a business objective.

3.3.5 MbC Architecture

The MbC [39] architecture extends the classic IT Management stack. At the bottom lies the Monitoring Layer which probes the liveliness of the service components and in particular system or service parameters. When a system is down, a failure alarm is generated. Consequently, a list of impacted services is determined. When comparing service parameter measurements to given thresholds extracted from SLAs and in cases of non compliance, violation and degradation alarms are also reported.

Both types of alarms are used as input to the second layer of the stack, namely the Diagnosis Layer. The purpose of this layer is to identify causes of faulty behavior.

Causes are then used as input to the third layer, the Recovery Planning Layer which seeks to determine recovery plans for the input causes. As a result of this analysis, multiple options describing recovery plans and associated cost are determined.

The Monitoring Layer also provides information such as violation events and proactive alarms (service degradation, etc.) to the SLA State Tracking Layer which keeps track of the current state of each SLA managed by the enterprise.

The fourth layer, the Contract-based Analysis Layer, informed by the SLA State Tracking Layer gives a business context to the options generated by the Recovery Planning Layer. Based on the analysis on the business engagements and objectives, captured within the SLAs that the enterprise committed to, the Contract-based Analysis Layer associates a utility value to each of the options. This utility reflects the overall impact that a recovery option would have on the use of the services impacted. It is obtained by analyzing the consequences of violating or complying with the various service level objectives agreed during the negotiation phase of the contract, and the costs associated with each option.

Finally, the Decision Making Layer chooses the option that maximizes the utility. The decision is then communicated to the Service and Monitoring Configuration system that applies the recovery plan associated with the decision.

3.4 Service Provisioning Work

The challenge for a service provider is to make the provisioning process most efficient. Provisioning systems thus have to be developed to support automation of this process effectively. In order to tackle this issue, different solutions have come up over the past years. Examples are shown of fundamentally different approaches in four categories.

3.4.1 Template-based Service Provisioning

[11] introduces a generic service provisioning approached based on profiles. Reusable parts are subsumed in a machine readable service profile that is then interpreted by a generic service engine. The provisioning process is specified in two tiers. The upper tier, the service layer, consist of the description of a service offering. This service profile shows a very abstract view of the service and is completely independent from any technology. At the lower tier, the network layer, generic service offerings are mapped onto particular network technologies. For this task, a provision profile describes how the service description is translated into configuration parameters. It specifies configuration templates that are applicable to the involved network devices, configuration delivery methods, as well as a set of task templates. The configuration template determines configuration files for particular network elements and the instantiation of a template specifies the concrete values in a configuration file. Dynamic aspects are defined in task templates that detail provisioning steps over multiple devices. With the aforementioned concepts, a formal specification of service provisioning is obtained. Second, a provisioning engine that interprets those descriptions has been developed.

Subramanian and Lewis propose a different system, in their approach the usage profile is automatically developed by monitoring the usage for a period. Based on this observation and the SLA, future allocation of resources is done continuously [43].

3.4.2 Model-based Service Provisioning

[18] presents a model-based approach for automated service provisioning of shared services. The foundation of the system are two model repositories representing the desired state and the observed state. Tools incrementally determine the differences between the repositories and apply changes in the environment. Both models use Common Information Model (CIM) and model elements can be used in both domains. The desired state model contains the descriptions of the services and the goals for provisioning the service, and it is filled by the operator. In contrast, the observed model represents the current state of the environment, and it is changed by executing actions in the physical environment. Tools read the output of other tools within the desired state repository and compare ist to the observed state, basing their actions thereupon. Execution steps are coordinated using a workflow description and the shared knowledge in the desired state repository.

3.4.3 Case-based Service Provisioning

The application of Case-Based Reasoning (CBR) techniques for service provisioning is proposed in [16] and especially addresses the question how to obtain the requested respectively desired granularity of fulfillment, operational and withdrawal steps. CBR offers solutions to a new problem based on past experience. Previously solved problems are gathered as cases in a case library and whenever a new problem arises, a similar case has to be found there. To apply this paradigm to service provisioning, the idea is to describe a service as a service template, including service ordering information, the necessary fulfillment and operational steps, the withdrawal steps if the service needs to be removed from the system as well as the policies associated with this service. Having such a service template structure which corresponds to a case structure, enables us to apply the CBR paradigm to dynamic service provisioning. The main idea is to think of a service template as a case. If a new service is ordered by a customer, then the service template library (case repository) is searched for a similar service order, respective service template. After a similar service template is found, it is retrieved and adapted to the specific customer

requests (specified in the service ordering), and afterwards the adapted steps are executed. The proposed application of the CBR paradigm successfully addresses the following problems: (i) to gain the requested granularity of steps (as precisely as possible) and (ii) to ensure up-to-date provisioning steps since these need to be updated due to changes, *e.g.*, in the infrastructure or in customer requests.

3.4.4 Agent-based Service Provisioning

[30] argues for an agent-based approach to service provisioning. ASPOSE (Agent-based Service Provisioning in Open Services Environment) facilitates service provisioning across multiple-domain networks and is aimed at context-aware mobile telecommunication services. For this purpose, four kinds of agents are defined, the user device agent represents the user terminal at the network side, the service agent encapsulates the service logic, the user interface agent deals with specific terminal requirements related to a service agent, and the terminal agent represents the terminal's capabilities. All services in the system are modelled as a service agent containing the service logic. During service deployment, the service agent provides an XML document to the different domains that contains the relevant information.

4 Multi-provider Models

In view of a world-wide network it is obvious that multiple networks will be present and they are interconnected in many different technologies and at many locations in the world. However, this situation leads to the key concern that many management tasks need to address the so-called multi-provider problem, which is separated into (a) the multi-domain dimension in technology terms, (b) the analysis of external effects in such multi-provider scenarios to identify the economic consequences of technology interconnection schemes, and (c) the multi-domain model and its service offerings across domain boundaries.

4.1 Definitions of Multi-provider Dimensions

The main technical issue in Multi-provider scenarios is the coordination and communication between the partners concerning all aspects involved. If services are to be provided by another operator, than the one the user originally subscribed to, in an integrated fashion (that is, the change of provider being transparent to the user), knowledge has to be shared between the two operators. The solutions for this should fulfill a set of requirements:

The mechanisms have to be privacy aware: This concerns both the security of communication channels between service components as well as the flexibility of mechanisms involved in service provisioning and accounting activities with respect to the set of shared customer and service details.

The communication should be efficient with respect to all resources (including bandwidth and time). For instance: different charging models require different subsets of measurable parameters in accounting. This implies either fully synchronized knowledge of all potential services/service variations between all operators or mechanisms for distribution and ad-hoc configuration of provisioning and accounting tasks. Also the scope of events should be limited as much as possible, involving only components directly involved. For instance: an intra-domain handover at a foreign provider should not be handled directly by the home provider.

4.1.1 Technical Aspects

Thus, in order to allow arbitrary cooperation in multi provider scenarios, a technical framework has to be defined with resect to the following aspects. These aspects are structures after the basic premises for cooperation: Each instance ought to know what it is supposed to do and the requirements therefore, it should know how to handle events, it should know how to contact other instances and be able to communicate with them in a secure manner. Thus, the following aspects (cf. Figure 2) should be covered by the derived Service Model (see Section 5.1.1):

Task distribution: A role model for both organizations as well as components has to be developed, of which responsibilities and informational requirements can be derived.

Interaction schemes: A mapping of events to routines has to be designed, which indicates, where and when interactions are supposed to take place and how events should be handled.

Inter-domain interface: Connection points have to be defined that allow arbitrary and ad-hoc communication between components of different operators.

Communication parameters: Protocols, extensions, formats and units or meta languages have to be agreed upon.

Security: Authentication and authorization as well as protections for communication privacy, integrity, availability etc. have to be in place in order to insure a correct operation.

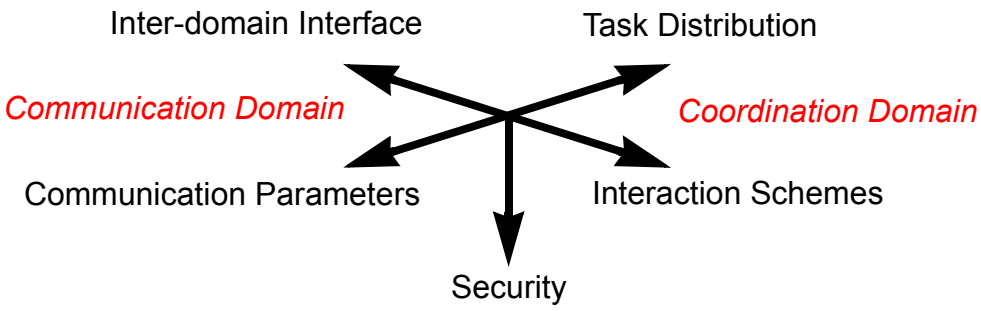


Figure 2: Technical Aspects

4.1.2 Economic Aspects

In economic terms, the cooperation has to be mutually beneficial to all partners. In order to achieve this, the following economic aspects (cf. Figure 3) have to be considered:

Economic role definition: based on investigations on the usage balance (one-directional, bi-directional, balanced, unbalanced) a relationship as peers or customer-provider has to be defined.

Cost analysis: a cost analysis over both direct (*e.g.*, equipment) and indirect (*e.g.*, impact on other services) cost factors has to be accomplished in order to determine reasonable compensation rates.

Cooperation angles: relationships concerning support cooperations or establishment of a common service portfolio in order to maximize synergy effects should be defined for efficient cooperation.

Definition of legal contracts and trust relationships: out-of-band conditions of the cooperation have to be specified in form of legal contracts, including specification of penalties upon failure of compliance and clarification of trust relationships.

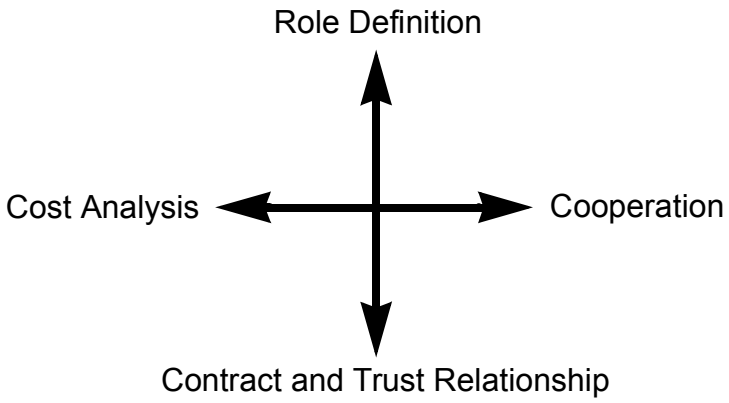


Figure 3: Economic Aspects

4.1.3 Operational Aspects

On the operational side (cf. Figure 4), a management basis has to be provided for the cooperation. The management basis should allow the sharing of necessary information, the synchronization of service portfolios as well as the supervision of cooperation performance. Thus, the following questions have to be covered:

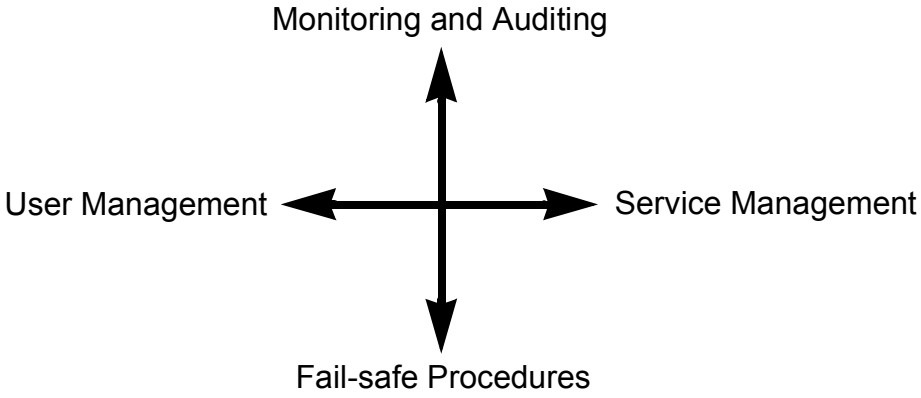


Figure 4: Operational Aspects

User management: how are user databases either synchronized or users of cooperation partners authenticated/authorized?

Service portfolio management: how are service details synchronized, mapped on each other in different domains and offered/maintained/accounted for? This might (*e.g.*) be leveraged by establishment of a common Service Catalogue (see Section 5.1.1).

Monitoring/auditing management: when is monitoring or auditing taking place? Where? In which form or on which parameters? Where are the results evaluated?

Fail-safe procedures: Which measures are in place to insure the availability of services or necessary processes (*e.g.*, accounting) in case of arbitrary component failures?

4.2 External Effects of Multi-provider Scenarios

Externalities constitute an important field for research in the domain of economics for decades. In particular, a number of seminal studies was published in the 1950s and 1960s. Nevertheless, externalities and their effects remain highly controversial topics today. This is mainly due to the fact that externalities are closely interrelated with one of the most

fundamental questions in macro economics—the question whether governments shall follow a so-called laissez-faire policy or intervene actively by means of regulatory acts.

Externality as a non-judgemental term subsumes both, external costs and benefits, whereas external costs are often used synonymously to externalities. In accordance with this tradition, externalities determine mostly external costs in the following. It has to be noted, however, that analogous conclusions are always possible to be drawn for external benefits.

Externalities result from a potential discrepancy between public (also referred to as social) and private costs [37]. When calculating costs and benefits of an undertaking, individuals and corporations tend to consider their own (so-called private) costs/benefits only. Thus, effects to third parties in terms of costs and benefits caused by such an undertaking are ignored. These effects are called externalities.

Hardin's renowned studies visualize the problems resulting from not internalizing external costs exemplarily for commons (commonly available goods). External costs that remain external lead to overuse of the common good—an effect of (Pareto-)inefficiency known today as the tragedy of the commons [25]. Such cases explain why externalities were often equated with market failure [4], whereas Pigou concluded from the existence of external costs that governments—perceived as outside agents, thus, also seen as third parties—should use taxes as a means to actively influence (*i.e.*, limit or increase) those activities of an party that causes externalities [37]. Governmental influence by imposing taxes is in direct conflict with a laissez-faire policy as supported by Adam Smith who propagated the concept of the "invisible hand", meaning that individuals that optimize according to their individual (as opposed to social) welfare would be "led by an invisible hand to promote [...] the public interest" [41].

The Coase theorem [13] proves that the existence of externalities is alone not adequate cause for governmental measures. Provided the following preconditions, external costs do not lead to inefficient outcomes—regardless whether the externality causer can be held liable or not:

Transaction costs are irrelevant, *i.e.*, absent.

Property rights of all involved stakeholders are defined.

Under such conditions, involved parties would always bargain for a Pareto-optimal situation where the externality causing party has the incentive to compensate the others for primarily externalized and ultimately internalized costs. This allows for the set of key conclusions to be drawn as follows:

Governments shall limit themselves to a role facilitating bargaining at a minimum cost.

Resource property rights shall be assigned to that party that can gain the highest profit from using the resource.

In case of uncompensated externalities, liability agreements lead to efficient outcomes.

Transaction costs are manifold as they consider any cost type encountered during a transaction that bases on a contractual agreement. Transaction costs do not include the costs for the good to exchanged between contractual parties, but they subsume *ex ante* costs, *i.e.*, costs incurred before a contract was concluded, and *ex post* costs, *i.e.*, costs incurred after a contract became effective. The first embraces costs for search and information and bargaining costs, namely contract negotiation costs. The latter covers policing and enforcement costs, which includes costs for processing, control and enforcement in case contractual terms were violated. Table 1 lists for two directly contract-

related (SLA negotiation and SLM) network management activities as well as for those five network management activities of the ISO (International Organization for Standardization) FCAPS (Fault, Configuration, Accounting, Policy, and Security Management) model the respective expected transaction cost weights.

Table 1: Expected Transaction Cost Weights ("X" main weight, "x" lower weight, "-" neglectable)

Transaction Cost	SLA Negotiation	Service Level Management	Fault Management (F)	Configuration Management (C)	Accounting Management (A)	Performance Management (P)	Security Management (S)
Ex ante (Search, information, bargaining)	X	-	x	X	x	X	X
Ex post (Policing, enforcement)	-	X	X	x	X	x	X

SLA negotiation directly reflects those *ex ante* transaction costs before a contractual agreement was concluded. Analogously, SLM marks *ex post* transaction costs as SLM is concerned with continuously monitoring those terms that were included in an SLA. Fault and accounting management activities are expected to be most costly after contract conclusion, in particular in processing phase (*e.g.*, costs for monitoring that accounting records are completely and correctly generated) and in enforcing rights when a fault has occurred. The latter holds also for security management, whereas transaction costs are expected of equal weight in contract negotiation phase when security goals need to be specified. Similarly, configuration and performance management activities are assumed to cause the highest transaction costs *ex ante*.

In a multi-provider scenario, absence of transaction costs is not easily assumed. In contrast, the higher the number of providers, the higher the chance for increased bargaining costs for a potential externality compensation needs to be accounted for. This is due to the fact of disproportionate growth in (direct) communications paths with an increasing provider number.

In this light, the question of correct resource property rights assignment lapses—it receives at least lower significance with regard to Coase's theorem and externalities. On the other hand, *liability assignments*—technically represented by the according SLAs in electronic contracts (*cf*. Section 5)—for potential uncompensated externalities obtain highest importance.

4.3 Multi-provider Model: Domains, Services

This section describes two additional service provisioning templates, in addition to the two existing ones of Section 4.2.3 of EMANICS Deliverable D8.1.

The first service provisioning template is a very generic one that makes a clear distinction between ASPs (Application-level Service Provider) and ICPs (IP Connectivity Provider) and tries to take into account all possible relationships in a multi-domain environment. That means that this service provisioning template can act as a generic template, from which more specific service provisioning templates (including the service provisioning templates of Section 4.2.3 of EMANICS Deliverable 8.1) for specific types of services can be derived.

The second one falls completely to the other end, being targeted towards a specific type of service, showing, however, the very specific roles that the relatively abstract stakeholders of the templates presented so far have to fulfill in order for this specific type of service to be offered.

With respect to the above, the former is the AGAVE-compliant service provisioning template [50] and the latter is the ENTHRONE-compliant service provisioning template [3].

4.3.1 Service Provisioning Template 3 — AGAVE-compliant

This service provisioning template is similar to the pure service provider-compliant template described in D8.1 of Emanics. The main difference is that in the AGAVE-compliant template, in addition to horizontal business agreements between ASPs (that are ASIAs) which are used to expand the scope of one type of service, vertical business agreements between ASPs (that are ASP service provisioning agreements – ASPAs) are taken into account. The role of the latter is not to expand the scope of one type of service, but to allow ASPs to enhance the services provided by other ASPs to provide a new type of service. ASPs are allowed to establish ASPAs amongst themselves, while customer stands for customer of other ASPs.

As an example of this scenario, one could consider the case where the customer ASP (as shown in Figure 5) establishes an ASPA with another ASP to use its Voice-over-IP service and on top of that provide an audio conference service to its own service customers.

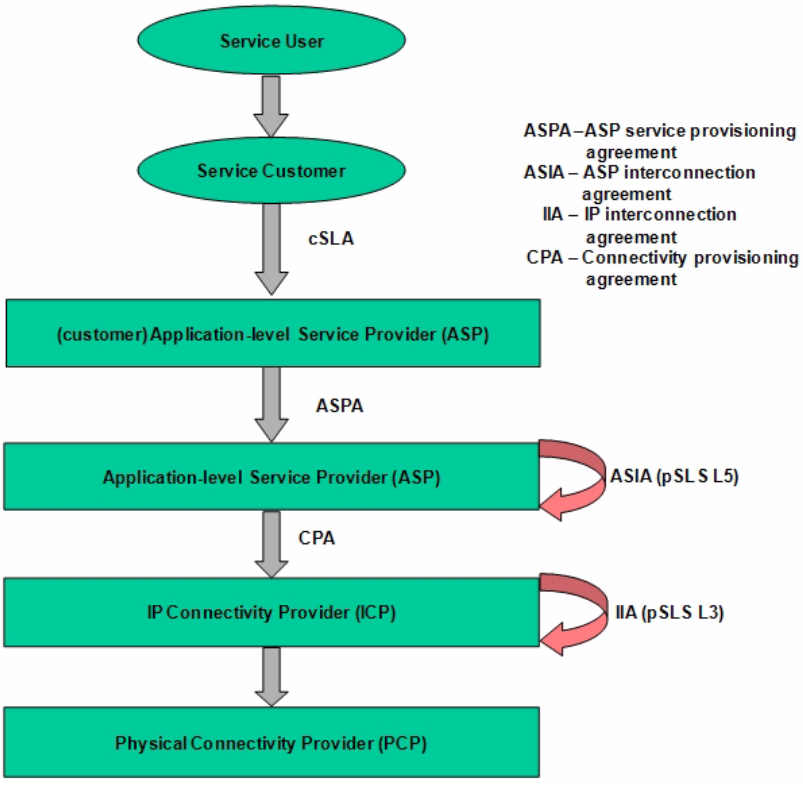


Figure 5: Service Provisioning Template 3 – AGAVE-compliant

4.3.2 Service Provisioning Template 4 — ENTHRONE-compliant

This service provisioning template is very much targeted towards QoS enabled delivery of multimedia content (*e.g.*, Video-on-Demand) to customers several domains away from the content server itself.

The template of ENTHRONE (cf. Figure 6[1]) aims to provide a complete solution by constructing an end-to-end content distribution chain and addressing the functions of preparation, adaptation and distribution of multimedia content to various end-user terminals. The roles and responsibilities of the actors involved in the end-to-end QoS chain are tightly coupled with the specific aim of this template, therefore names of actors are kept as given in [3].

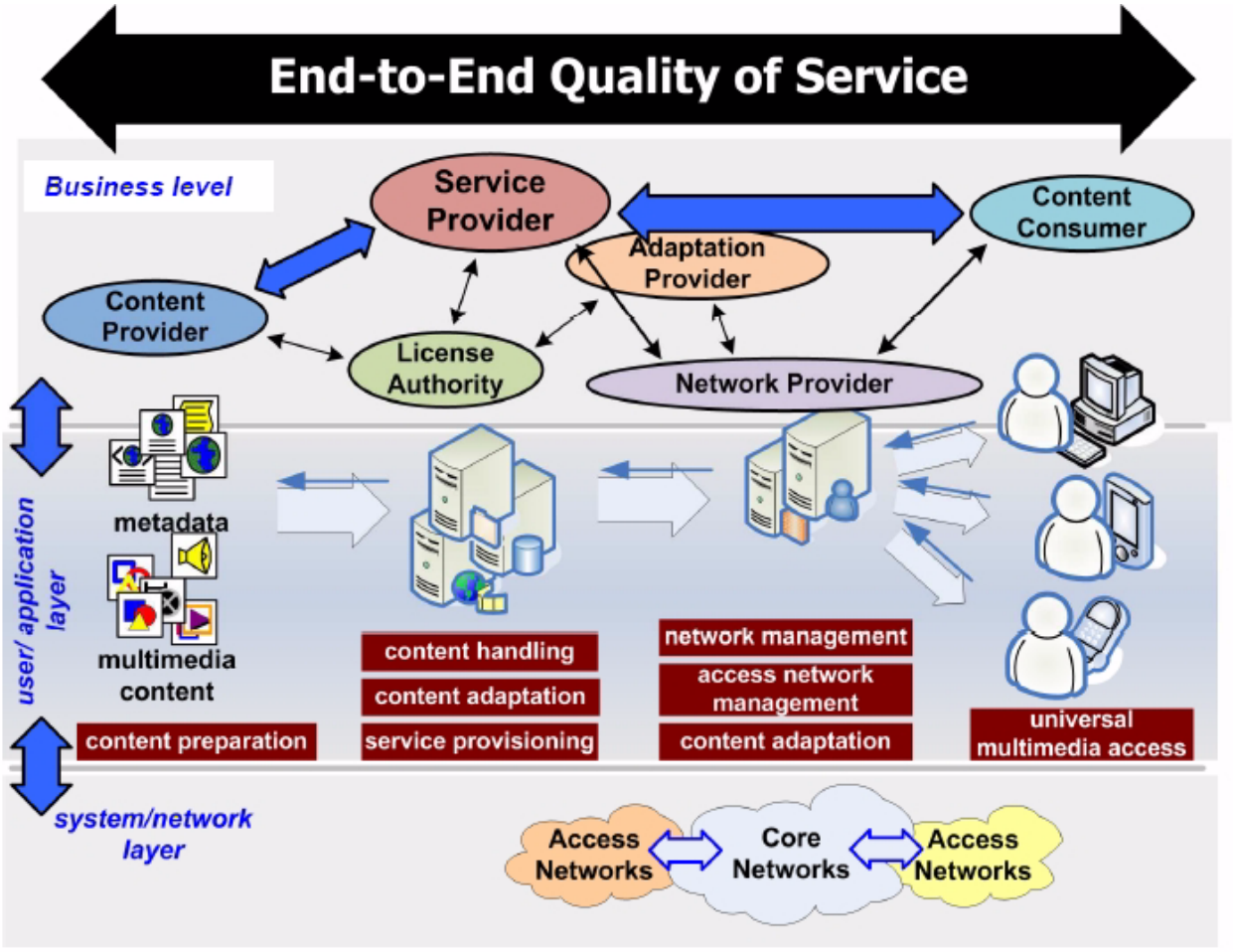


Figure 6: Service Provisioning Template 4 – ENTHRONE-compliant (cf. Footnote 1)

The main actors and their respective functions along the distribution chain are as follows:

The Content Provider (CP) prepares the actual multimedia content as MPEG-21 Digital Items (Motion Pictures Expert Group), facilitating scalable coding formats and meta-data formats providing means for:

- Indexing, searching and retrieval and
- Content negotiation, transaction, re-proposing, adaptation, and personalization.

In general, CP owns content servers for storing the multimedia content.

The License Authority operates in close relationship with the CP or the Service Provider. It provides digital rights expression and management.

1.This figure was taken from the Enthrone web site; the missing arrows between Adaptation Provider and Service Provider as well as between other entities exist because exact relationships in this template are still under development and not finalized. This figure reflects the current state at the time of writing.

The Service Provider (SP) provisions and offers multimedia services to end-users and enriches the multimedia content with additional meta-data with respect to SLAs taking into account constraints imposed by access networks for service provisioning towards the customer:

- Content handling, i.e., content aggregation, adaptation, and edition (*e.g.*, adding dynamic properties to the "static" Digital Items received from the Content Providers for streaming over heterogeneous networks);
- Management of the usage context, i.e., user preferences, terminal capabilities, network characteristics, and the natural environment; and
- Service provisioning (through cSLAs with end-users and pSLAs with the Network Provider where the CP is physically connected to).

The Adaptation Provider operates in close relationship with the Service Provider, the Network Provider(s) and licence and certification authorities. Its goal is to provide improved QoS of content delivery while optimizing available system and network resources across the end-to-end chain. It takes content adaptation decisions according to the a-priori-known as well as dynamically received context information. Content adaptation is done by the Service and/or Network Providers.

The Network Provider (NP) offers QoS-based IP/DVB connectivity services at its network autonomous domain level. Co-operation is needed among network providers for providing inter-domain QoS-based IP/DVB connectivity services; this is done by establishing pSLAs with the neighboring NPs in order to extend the QoS-chain till the end-customers. Both core and access networks are owned and managed by NPs.

The Content Consumer requests through a cSLA the services provided by the Service Provider and consumes it through his end-device. The end device functions as well as all equipment along the end to end delivery chain are MPEG-21 compliant to implement Universal Media Access (UMA):

- Smooth rendering of scalable content while enforcing digital rights imposed by individual actors within the delivery chain, i.e., content, service, and network providers, and
- Context information management, generation and transmission towards the Service Provider.

5 Service Level Management

Service Level Management (SLM) is one of the most important management disciplines in IT Service Management (ITSM) [34], [35], vital for customer-orientation and provision of high-quality IT services. SLM is responsible for determining, monitoring and reporting IT service quality metrics (QoS parameters) in line with the economic goals of the entire organization. It is important for an improved relationship between a service provider and its customers, because a common understanding of expectations and possible achievements to the agreed costs is established between provider and customer.

In a multi-domain context, SLM needs to face several additional challenges some of which are also addressed in this section. In general, it can be said that SLM in a multi-domain scenario will be a research focus in the future of the EMANICS project.

5.1 Organizational Frameworks for SLM

Various concepts for supporting effective Service Level Management have evolved from research and practice throughout the last couple of years. These approaches differ strongly in their scope, their level of detail, their feasibility for technical and tool support and their target audience. This heterogeneity may turn out as problematic when IT managers try to implement SLM: On the one hand, a holistic approach does not exist, and on the other hand the existing concepts, frameworks and technologies do not fit together like pieces of a puzzle. Guidance in integrating multiple efforts into one consistent SLM solution suite often is not available today.

The goal of this section is to shortly introduce the most important approaches ("frameworks") that support Service Level Management from primary an organizational perspective. Two of the probably most popular existing SLM frameworks are the IT Infrastructure Library (ITIL) [34], [35], and the NGOSS SLA Management Handbook [45]. Both frameworks claim for themselves to be business-aligned. While the NGOSS Handbook is clearly focused on SLM issues only, ITIL is not. In fact, ITIL provides guidelines (best practices) for the entire field of ITSM. Service Level Management is one of the five reference processes described in the Service Delivery book.

5.1.1 Foundations and Terminology

In the area of SLM, various terms and concepts have been established over years and are today shared between different approaches. Although—as almost everywhere—a uniform terminology for SLM does not exist, the following set of terms is used in the majority of the presented approaches in basically comparable meanings.

Service Level Agreement (SLA): A Service Level Agreement is a written contract between a service provider and a service customer/subscriber. It must contain a description of the service functionality, definitions of related QoS parameters (service levels) and declarations of responsibilities of both parties. It may additionally contain prices for service usage to pay by the customer/subscriber and penalties for service level violations to pay by the service provider.

Service Catalog: A Service Catalog contains definitions of standard services as well as documentations of customer-specific services. It can be used as a foundation for automated service subscription or for the negotiation of SLAs.

Service Model, Life Cycle and Domains: When talking about SLM and SLAs, there should exist a common understanding of what the term service means. A view on a service consists of two components: the service life cycle which displays the dynamic behavior of a service, and the static service model [20], [21] which describes the composition of basically entities and interactions inside a service and shows the service in a role-based context.

- Starting with the service life cycle model, a division of the life cycle into seven phases has proved as a reasonable scheme. These phases are: offer, negotiate, implement and test, accept, operate, change and decompose. Please refer to D8.1 [42] for a more detailed description.
- The relevant domains for the service context in the static service model are the provider domain and the customer domain. The provider domain comprises all of the entities vital for providing the specified service functionality. The service provider is responsible for the task of service provisioning and therefore operates a service implementation and a service management. The customer domain contains the

customer and the user role. The user can deploy the usage functionality of the provided service via a Service Client (SC) which is connected to a Service Access Point (SAP). The customer subscribes the service, concludes an SLA with the service provider and monitors service provisioning via the Customer Service Management (CSM) access point. Further on, the model defines functionality classes and several interfaces for management and usage. Please refer to D8.1 [42] for a more detailed description.

Figure 7 is a simplified depiction of the service model defined in [20], focused on the illustration of service provisioning in a multi provider environment. This model can be regarded as a simplistic foundation for the following subsections.

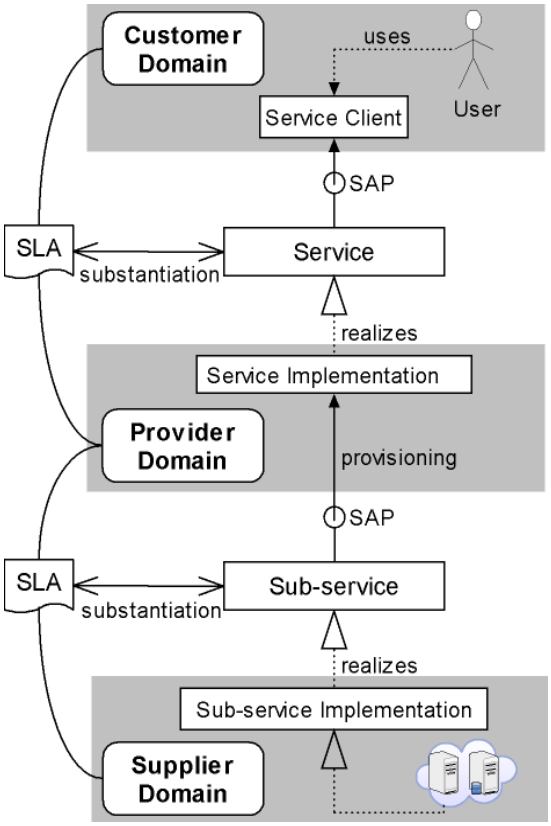


Figure 7: Simplified IT Service Model with Multiple Domains and Service Chaining

5.1.2 Analysis of Frameworks

In the following, a survey is given by addressing ITIL SLM and the NGOSS SLA Management Handbook.

5.1.2.1 IT Infrastructure Library (ITIL)

ITIL (IT Infrastructure Library) is a collection of books, in which best practices in IT Service Management (ITSM) are described. Today, ITIL can be seen as a de-facto standard in the discipline of ITSM, for which it provides guidelines by its current core titles Service Support [34] and Service Delivery [35].

ITIL follows the principle of process-oriented (IT Service-) Management: Every management activity taking place within an IT organization is part of one of the defined management processes. Thus, process-orientation extends the idea of functional management where IT management decisions and actions take place in different

departments (*e.g.*, network department, server department, storage department). In effect, the responsibilities for specific IT management decisions can be shared between different organizational units as the management processes span the entire IT organization independent from its organizational partition.

Service Level Management is one of the ten ITSM processes defined by ITIL and part of the Service Delivery book. Besides Service Level Management, this book describes Availability Management, Capacity Management, IT Service Continuity Management and Financial Management for IT Services. Together, these five processes are called the tactical processes and build the direct context of SLM—in contrast to the operational ITSM processes described in the Service Support book (*e.g.*, Incident Management, Change Management).

The relevant roles in ITIL Service Level Management are the service provider, the service customer and the service user which exactly maps the generic service model introduced in Section 5.1.1. The SLM process builds the interface between the IT organization (as the service provider) and its internal and external customers. According to ITIL, any customer is characterized through the commission, payment and ownership of one or more IT services that are provided by the IT organization. Due to the roots of ITIL in the British government, the focus is clearly set on internal—which basically means non-commercial—customers like the manufacturing or accounting department of the enterprise in which the IT organization is located. A user is defined as any person using a commissioned service (*e.g.*, manufacturing staff or an employee in the accounting department).

SLA Types and Structures

ITIL enhances the SLA concept by some additional aspects and terms: Any SLA that is closed between the IT organization and an internal sub-provider is called an Operational Level Agreement (OLA). By contrast, an SLA that the IT organization contracts with an external supplier in order to obtain (sub)services from this provider, is called an Underpinning Contract (UC). ITIL makes this distinction, because OLAs and UCs may significantly impact what can be promised within an SLA. The Service Level Manager must know about the interdependencies between the existing UCs, OLAs and SLAs.

Besides these two special types of SLAs, ITIL defines three different kinds of SLA structures. An SLA is called a Service-based SLA when it is valid for all customers of one or more services and no individual SLAs are designed for different customers. This is often the case for services facing uniform customer/user requirements (*e.g.*, e-mail, internet-access). By contrast, a Customer-based SLA is closed with one individual customer and normally for all the services subscribed by this customer. Customer-based SLAs basically find appliance when customer demands regarding service quality parameters vary to a high extent. The third SLA structure ITIL proposes is the so called Multi-level SLA which can be regarded as the composition of Service- and Customer-based SLAs. ITIL describes Multi-level SLAs to follow a three-layer structure:

Corporate Level covers generic issues valid for all customers (*e.g.*, opening hours of the Service Desk).

The Customer Level contains customer-specific issues as extensions to the Corporate Level, regardless of the services ordered by this customer (*e.g.*, expanded Service Desk opening hours for some customers).

The Service Level covers service-specific issues relevant for one customer or group of customers (*e.g.*, agreed availability for an accounting service).

The idea behind this three-layer structure is to substitute only the service and customer layers in order to avoid redundancy and reduce maintenance and administration expense when designing new SLAs.

The Management Process

There are six activities that build the ITIL Service Level Management process: Identify, Define, Contract, Monitor, Report and Review. Each of these activities needs certain inputs and creates certain outputs both of which are described by ITIL.

Identify: First of all, the customer demands have to be identified and described as Service Level Requirements (SLRs). This document will provide a basis for the future SLA.

Define: In the second step, the concrete service which is to deliver to the customer, has to be defined in case of a new/individual service. Therefore, a Service Specification Sheet (Specsheet) should be created, containing information on the technical implementation and realization of the service. The Specsheet can be seen as a translation of the SLRs into technical specifications. Additionally, ITIL proposes to develop a Service Quality Plan (SQP) as an internal document containing i.a. key performance indicators and a plan for achieving the agreed service quality. In case of existing services, the respective documents can be adopted.

Contract: In this activity, a written contract between the IT organization and its customer is closed, based on SLR, Specsheet and SQP. Another output of this step is the updated Service Catalogue. In case of a new or modified service, the changes should be added to the Service Catalogue in order to potentially provide this service to other customers, too. The Contract activity also implies to close or update UCs with external suppliers, if the agreed service requires sub-services, technology or infrastructure which the IT organization is unable or unwilling to provide by itself.

Monitor: Of course, the actually achieved service levels (QoS parameters) have to be monitored. It is important that the SLA contains information on how and how often the measurements have to take place. Measurement outputs are called Service Level Achievements and serve as inputs for the next activity.

Report: In this activity, the achieved service levels are compared with the agreed service levels in order to detect violations. Service Level Reports are created and handed to the Service Level Manager.

Review: As the last activity in the ITIL SLM process, the Service Level Reports are evaluated with respect to the contracted SLAs, OLAs and UCs and of course under consideration of the SQP. In order to continuously improve the service quality, a Service Improvement Program (SIP) should be developed and launched within the next time period.

ITIL gives guidance for the organizational setup of an SLM process. It is focused on a clear role model, the definition of responsibilities, activities and a common terminology as well as business-awareness and -alignment of ITSM. Covering this radius, it remains superficial in many areas. It is tool-independent, only little technology-aware and does not provide concrete templates for the various artifacts (even not for an SLA).

5.1.2.2 Next Generation Operation Support Software (NGOSS)

The SLA Management Handbook is a publication of the Tele Management Forum (TMF) and part of the New Generation Operation Support Software (NGOSS) project. It consists of four volumes named Executive Overview [45], Concepts and Principles [46], Service and

Technology Examples [47] and Enterprise Perspective. The fourth volume (Enterprise Perspective) has not been released to public yet.

Although the contents of the NGOSS Handbook are basically aligned to businesses in the telecommunication industry, they are generally also applicable to a broader scope of IT-dependent organizations, because most of the presented concepts are not restricted to telecoms industries specifics.

The probably most significant title of the SLA Management Handbook suite is the second one (Concepts and Principles). It starts with some Business Considerations including a business model, followed by a chapter on Telecommunications Services. The next three chapters deal with several subjects directly concerning SLAs, starting with SLA Content and Management, followed by SLA Management Tools and concluding with a chapter on SLA Performance Reporting. In the following, a short survey is given on contents of these five core chapters.

Business Considerations

In contrast to ITIL, the role model applied in the NGOSS Handbook is more diversified, particularly with respect to the existence of external suppliers in the end-to-end service delivery chain. But the basic business model is the same as already seen in ITIL: Basically, an SLA is regarded as a contract between one customer and one service provider. This contract is located at the customer-provider interface. An extension of this simple model is the provider centric Business Relationship Model. It adds the roles of complementary providers, third party service providers, function and process suppliers, intermediaries and hardware/software/solution vendors.

As the biggest stumbling block in SLM, the NGOSS handbook describes the end-to-end service challenge that consists in the delivery of a seamless service through a number of trading partners which is indistinguishable from the same service provided by a single supplier. In the remainder of the handbook, this e2e service challenge is always kept in mind. The handbook even claims to be the first open document addressing e2e service performance issues in SLM.

The Business Considerations chapter finishes with a section on service measurement and performance metrics. It first defines four basic prerequisites for an effective service measurement:

Parameters must be measurable.

The quantification method must be described.

A review of the delivered performance must take place.

Penalties or incentives must be specified.

Furthermore, a metric is defined as a measurable parameter. Ten requirements are defined that should be fulfilled by any specified performance metric. To give an impression, three exemplary name of them are as follows:

The metric should provide concrete repeatable measurements in well-defined quantities and without subjective interpretation.

The metric should be useful as a specification in a contract in order to enable the customer purchasing the service level he needs.

The measuring process should be acceptable to service providers and customers, and artificial performance goals should be avoided.

Implicitly, all of these business considerations build the requirements for the developed solutions and draw the context for them.

Telecommunication Services

A telecommunications service is characterized by an object-oriented model with the two central entities Service Function and Service Resource. The latter is a superclass of hard- and software, staff and intellectual property and licenses, demanding only hard- and software as mandatory components. Service Resources however enable Service Functions which are divided into the service's Primary Function, Enabling Function and OAM Function (Operation, Administration, Maintenance).

This model is much simpler than the generic service model introduced in Section 5.1.1, but not contradictory to it. It uses the same domains (customer and provider domain) and shares the concept of an SAP. New aspects are the possibility of aggregating SAPs to SAP groups and regarding different layered provider domains. An additional feature of the model is given by its Service Elements (SEs) that are abstract entities out of which a service is composed. In this capability, SEs can be shared between different provider domains, *e.g.*, a service provided by A may base upon an SE out of the provider domain of B although not being completely dependent of one of B’s full services. Thus, an SE can be a sub-service, a physical resource, a human resource or what ever may be used to build a service.

Furthermore, the NGOSS Handbook proposes a Customer Contact Point Reference (CCPR) model which serves as a model for exchanging service performance and management information between a provider and its customer. The Customer Contact Point (CCP) is the logical point at which a customer may manage the services he has subscribed for. In the generic service model of Section 5.1.2.2, the CCP is given by the CSM access point which can be accessed by the customer through a CSM client. Again, the two models are very similar.

With respect to service performance, the NGOSS handbook introduces additional concepts two of which are the Service Degradation Factor (SDF) and SAP Weighting. An SDF can be used when calculating service (un)availability. It is based on the idea that besides performing well and failing completely, a service may also be partially degraded, but still usable. In order to consider this fact in availability calculations, an SDF – which can take any value between 0 (service fully available) and 1 (service fully unavailable) – is assigned to each outage event type. Weighted SAPs can additionally be useful in order to take into account different impacts of outages related to different SAPs when calculating service availability.

SLA Content and Management

Based on the business considerations and the service model, the handbook gives recommendations for the concrete SLA content and design. These recommendations are arranged in four categories which are:

Fulfillment Process (Recommendations 1 to 5): This category contains recommendations concerned with the negotiation and engineering of SLAs.

Assurance Process} (Recommendations 6 to 13): After concluding the SLA, the recommendations in this category should be followed by the provider when delivering the service to its customer.

Customer Interface Management} (Recommendations 14 to 16): This category contains recommendations for the communication between provider and customer concerning SLAs and services.

General Recommendations} (Recommendations 17 to 23): The last category contains general recommendations viable for SLM like modular assembly of the SLAs or the definition of provider and customer responsibilities.

SLA Management Tools

The title SLA Management Tools' might mislead to the assumption that this chapter deals with software tools for SLM—which is not the case. It rather presents a Service Life Cycle model which is almost identical to the one outlined in Section 5.1.1, followed by a KQI (Key Quality Indicator) Development Methodology that shall help to identify metrics that capture the customer's QoS perception. The third tool is the SLA Parameter Framework that organizes performance parameters into specific categories.

SLA Performance Reporting

In the SLA Performance Reporting chapter, a Performance Reporting Interface, several sequence diagrams for different reporting scenarios as well as a Performance Reporting Process State Model are presented.

The NGOSS SLA Management Handbook covers much more aspects and detailed proposals vital for SLM than ITIL does. This is not surprising, since SLM is only one of a total of ten ITIL processes. While the second Volume of the NGOSS Handbook already consists of 204 pages of text, ITIL SLM is described within 33 pages in a comparable style. The descriptions in the NGOSS Handbook are less superficial and much more aimed at straight deployment.

5.2 Modeling and Formalization Approaches

The goal of this section is to shortly introduce the most important concepts that support SLA modeling and formalization from a more technical perspective than in the previous section.

5.2.1 Analysis of Mechanisms

There are mainly three types of technical mechanisms in SLM:

Service Level Negotiation

Service Level Measurement

Assessing the monetary business impact of Service Level Violations

These three technical mechanisms can be seen, in this order, as the logical steps covering the life span of a service, from its introduction till its actual deployment, and each of those categories of technical SLM mechanisms will be addressed in the following.

5.2.1.1 Service Level Negotiation

With the prospect of becoming the all-service network of the future, the Internet needs to provide services such as e-commerce, real-time voice and video applications, multimedia services, and so on. Such services require a minimum guaranteed Quality-of-Service (QoS). For the provisioning of QoS in such a QoS oriented Internet, when an Internet user/ customer wants to use QoS to support some service, they have to negotiate a Service Level Agreement (SLA) with the Application-level Service Providers (ASP) to request through the technical part of the SLA, that is the Service Level Specification (SLS), the resources and the associated QoS guarantees before they can actually use the service.

This negotiation process, however, in many cases is not automated or dynamic. The service level agreement (SLA) is usually agreed, either verbally or in writing, by both the

user and the ASP when a user signs up for a service. To change the SLA, normally a user has to contact and negotiate with the authority of the ASP, which then manually changes it. Compared to manual service negotiation methods, dynamic service negotiation offers higher degrees of flexibility to customers and the ASPs by reducing their time to request or to gain access to services. Recognizing the benefits of dynamic service negotiation, several protocols have been developed by researchers towards this objective [33].

The remainder of this section presents a summarizing table of various service negotiation protocols outlined in Section 3.3.3, allowing for the relative performance comparison of these protocols.

Table 2 gives a comparison of the aforementioned protocols (cf. Section 3.3.3) in terms of whether they allow for:

- SLS form transparency (beneficial, since the form of an SLS is not yet standardized)
- Extension of existing protocol (beneficial, since it means that they don't require extra functionality on top of existing protocols in order to be implemented)
- Reduced signalling (beneficial, especially for mobile devices)
- Lightweight operation (beneficial, especially for mobile devices)

Table 2: Comparison of Service Negotiation Protocols

Feature	RNAP	SrNP	COPS-SLS	DSNP	QoS-NSLP	QoS-GSLP
SLS form Transparency	No	Yes	No	No	Maybe	No
Extends Existing Protocol	Yes	Yes	Yes	No	Yes	Yes
Reduced Signalling	No	Maybe	Yes	Yes	No	Maybe
Lightweigtht	No	No	No	Yes	No	Yes

5.2.1.2 Performance Measurement

The next logical step after the negotiation and agreement of a service between a customer and an ASP (or organization in general) is the actual deployment of the service with the QoS requirements and clauses agreed in the SLA. In order for the organization to define and measure its progress towards its goals when introducing the service, relevant metrics need to be defined and deployed. Furthermore, in order for the organization to be able to verify that the service offered indeed matches the one defined in the SLAs so as to apply remedial actions in case of service degradation, SLA monitoring has to be in place during the SLAs life span.

Regarding SLA monitoring, the parameters that need to be monitored can vary greatly depending on the service to be offered and also the monitoring system is strongly influenced by the underlying infrastructure and the types of services offered, The generic notion of KPIs and the IBM's WSLA framework for SLA monitoring, as described in Section 3.3.4, are representative examples of how performance metrics can be defined and deployed and also of how SLA monitoring can be performed.

5.2.1.3 SLM Business Alignment

The last point mentioned in the IBM's WSLA framework brings into focus an important issue, that remedial actions and management actions in general should not contradict business objectives; in other words the need for SLM business alignment. Towards this

direction, HP has proposed a solution [39] under the name Management by Contract (MbC). MbC formalizes and analyzes contractual relationships in order to better inform the decision making process that underlies IT management. In brief, MbC is about assessing available options for IT management in light of the business impact they may have, as derived from contracts and SLAs. Available management options are compared with one another and ranked according to expected utility measures and the one maximizing the utility for the business (which may not be the most obvious) is selected. A more detailed description of MbC has already been given in Section 3.3.5.

5.2.2 Analysis of Models

A *model* is a approximate abstraction that offers a suitably idealized facsimile of a phenomenon. It approximates the salient features and hopefully allows one to understand the main factors at work and to make predictions. Modelling is a key strategy in science for extracting general truths from specific examples.

A model is distinct from a *framework*. Frameworks are technological toolkits generally used for implementing an idea by providing a standard set of tools. Models on the other hand are pedagogical, and scientific aids—they help us to think about the ideas, classify them and understand them. A modelling framework is thus a set of tools for building and models.

Models are usually classified by what basic assumptions they make and what features of a phenomenon they choose to suppress in the interest of clarity. Examples of modelling frameworks for service management, *i.e.* technologies for helping us to think about problems, are:

UML activity diagrams—object oriented ontology

Petri nets—or event-like process algebras

Promise theory—service oriented

5.2.2.1 Modelling Foundations

As a foundation for the analysis of models and modelling techniques, one can abstract from the IT service definition as given in 5.1.1 by saying that an IT service is generally thought of as an agreement to perform a function by one human organization for another in the context of information technology. It might be delivered through an underlying infrastructure “end-to-end”. The essence of the idea of service provisioning is summarized by a simple sentence which is referred to as an *abstract service*: A service is the performance of a potentially valuable function by one party for another. This definition is deliberately stripped of all specialist connotations. It can serve as a top-level ontological primitive for services.

In the present context of commercial broadband network services, special attention attaches to services that are provided by human business enterprises for money. Economic notions such as payment and customer guarantees then enter the scope of modelling. Quality of Service is a commonly used phrase. It is closely associated with the idea of a service level. Basic understanding of these notions are as follows:

A service level is a measurable rate of execution of the service function as observed by the recipient of the service.

Service quality is an aggregate metric which allows a partial ordering of service implementations. The *quality* of a service is a heuristic term whose precise meaning is defined by a policy agreed between the provider and the recipient.

The existence of quantifiable *service levels* already implies a model for services that involves the ability to quantify and measure activity or performance over time. To be able to

do this, time must be measurable and performed-service must be metered. In all cases for computer-centric services, service is actually countable. One measures service in terms of discrete transactions such as replies, packets or bytes. As human issues are added to the mix, it is possible to define a continuum of service, *e.g.,* in terms of continuum metrics like *availability*. See [8] for a discussion of continuous and discrete models.

UML does not not help us to model transaction frequency, only the structure of the protocols used for transaction. Petri nets are more suitable for handling discrete transactions, but the level of detail is too high (or the level too low) in general for practical modelling of services in an economics framework. Promise theory and contract modelling both position themselves at a level suitable for scaling aggregate value such as service delivery over an interval of time.

A definition for Service Level Agreement has already been introduced in Section 5.1.1. There is no agreed way of modelling Service Level Agreements today. None of the frameworks presented above gives concrete guidance in this direction. Some methods that have been used include Deontic logic, contract theory and promise theory (see below).

A contract is generally understood to be a bilateral collection of promises between two or more parties. The sum-collection of these promises then forms a contract specification which the parties agree to obey. Only when all parties have agreed (usually by signing the contract) is the specification a meaningful representation of the expected behavior of the client-provider relationship.

The only modelling framework currently capable of capturing this process seems to be promise theory, though the application details remain to be presented.

5.2.2.2 Modelling Service Management

In looking for ways to model service management issues needs to be discussed, such as:

The logical consistency of conditions required for delivery

The deployment of resources

End to end logistics of delivery in a supply chain

The estimated rate of delivery

The reliability of the delivery

The economics of delivery

Logical Consistency

Deontic logic (the logic of obligations) has been proposed as a way of discussing the logical consistency of managed services. Deontic logic has a long history as a supposed tool for discussing management and governance, though its achievements are debatable. It is commonly used to describe legal aspects of systems and commerce. Approaches using deontic logics for service management attempt to describe systems in terms of properties that they "ought to" have. If one assumes that it is possible to implement these requirements, it is possible to prove certain properties of the system. One of the principal impediments to this is the fact that individual parties in the system have different world views and might not even interpret requirements in the same way. See [12] for more information about modal logics.

Extended notions of modalities, such as temporal logics, process algebras, Petri nets and event calculi have also been used to discuss whether design-features, special states and properties can become true of systems and when.

A central weakness of logical approaches in these cases is that it is only a "destructive test", somewhat like stress-testing a bridge to failure. Logic can tell us if a series of relationships is insufficient, but it gives no assistance in designing these relationships. Moreover, even if it is possible to prove that a system will have a particular property, this proof is subject to unreliabilities which can invalidate the analysis. Logic is only a rewriting of certain underlying assumptions, but this has nothing to do with the real world. Observations need to be accounted for as well.

Reliability Theory

In order to bring the scientific method into modelling of computer systems and services, one must have a role for observation of reality, *i.e.* measurement. Reliability and fault analysis is a scientific engineering discipline that was developed significantly by the nuclear industry, where faults have high risk impacts. It comprises a wide range of techniques both deterministic and probabilistic to discuss whether systems will behave according to specifications. Reliability theory admits the important reality that systems are not completely reliable or predictable. Forces beyond the control of the system designer exist, and it becomes the responsibility of the designer to account for this probabilistic behavior in the design of the system. Reliability theory is a huge topic. See [26] for more details.

Network Effects

The approaches above tend to focus mainly on the roles of individual parts of a system, but the way the components are connected clearly plays an equally important role, especially in networking. Petri nets [15], social networking theory (centrality measures of important), and workflow models need to be considered [8].

Games, Decision Theory, and Contracts

Service delivery is not only a technical challenge but an economic one. Whether the economics is about money as in paid services or about abstract trade as in provider peering agreements, the value to the recipient of a service is a main motivation for the service provision. Economic modelling has many forms, but the theory of games and economics is a key ingredient. Game theory and its application domain "contract theory" are optimization methodologies for decision-making. There has recently been a revival of interest in the use of so-called rational decision theory in IT service management research. The annual BDIM workshop was started at NOMS2006 building on this idea. For more information see [39], [53], and [7].

Promise Agreements and Voluntary Cooperation

Promise theory was introduced by the Oslo EMANICS group in 2004 (first published in 2005) as a way of combining all of the key aspects of the above modelling tools into a unified approach. It makes a basic change of philosophy, removing the notion of obligation entirely and replacing it by the idea of voluntary cooperation. Since obligation can be described in terms of voluntary cooperation (but not vice versa) this viewpoint is more fundamental. It is also a constructionist theory, not just a stress-testing framework. See [9] and [36].

Promise theory attempts to combine the useful aspects of each of the foregoing approaches. It is a way of modelling networks of agents cooperating in an arbitrary fashion and performing abstract services for one another. The idea is to imagine that all of the activities in a system are promised voluntarily. This special viewpoint implies that all of the details of cooperation must be made explicit in cases where parties have obligations. An

advantage here is that no hidden assumptions can exist, and this leads to a great simplifications of the combinatorics of promises over logic-based approaches. The key features of promise theory are

There exists a collection of agents.

Agents are autonomous. They make their own decisions and no agent can be forced to perform any function by another. (Actions and decisions made by an agent are entirely voluntary.)

Each agent has private knowledge. It can share its knowledge with another agent by promising to reveal it (in the manner of a service).

There are two kinds of promises called + and -, or "service" and "use" promises. A + promise is a promise to give or provide some kind of abstract service. A - service is a promise to make use of an abstract service (if available)

A common mistake is to think of promises as communication transactions, rather than as abstract behavioral specifiers. A promise says nothing about the details of what is communicated between agents. One writes a promise as a directed graph from one agent to another.

$$A_1 \xrightarrow{b} A_2$$

b is called the promise "body" and contains details of what is being promised. An agent can only promise something about its own behavior, not about a third party.

A reliable binding between two hosts requires both a promise to serve and a promise to use the service.

$$A_1 \xrightarrow{+b} A_2$$

$$A_1 \xrightarrow{-b} A_2$$

A service promise is a key ingredient in a service level agreement. It is not strictly necessary for a client or consumer to promise to use the provided service, but it could be important to promise to use data or information from a party as part of the information exchange. For instance, open access control lists can be modelled as promises to accept certain data that are sent by the other party.

A service level agreement covers many different issues, with different promise bodies b_1,b_2, ... Some of these promises might be conditional on behavior from the other party, or on third parties. The collection of all promises relating to the interaction of the parties is a static document which ought to be logically self-consistent. Moreover, each promise can be evaluated for

its reliability or our trust estimate

its economic value to either party

its probable service level

Note that since time is not mentioned explicitly in promise theory, at least in the present formulation, it enters only as an implicit measure of experience in the form of probabilities such as reliability or expectation of service level. This is a realistic approach for real world applications where accurate data are seldom available and agreements are based on assumptions and estimates about delivery rather than observational facts.

There are many ways of using the promise graph to examine service quality and the inevitable uncertainties introduced both by the parties and the connections between them. This is an important tool for judging the likelihood of a service level agreement being kept.

Promise bodies contain both types and constraints. They describe relationships between multiple agents or parties, each with their own knowledge and “world views”. The logic of these constraints can be examined using straightforward first order logic once the promise graph is known. Most importantly there is a straightforward recipe for generating this graph and using the approach as a constructive modelling tool.

1 Identify the promises that are needed to address the needs of a recipient.
2 Identify the agents that are necessary to implement these promises, bearing in mind that each intermediate agent in a chain can alter the flow of information and service.
3 Evaluate the reliability and economic value of each promise to each agent to see the value chain.

5.2.3 Description of Relevant Interfaces to Multi-provider and Provisioning Concepts

As a vital discipline for the delivery of high quality IT services, Service Level Management needs to consider the context in which it takes place. To address this topic, two different perspectives are introduced and they show how SLM correlates to the previous as well as the next section of this document.

5.2.3.1 SLM in Multi-provider Scenarios

A cooperation of multiple service providers can take place in various forms. It can vary from a strict hierarchy with a single provider on the top to a pure heterarchic form where all providers are having equal rights. In the first case, a single provider offers a service to the customer and at the same time contracts the other involved providers to deliver sub-services. In the latter case, there is no superordinate organization and the providers coordinate their activities. The German research network provider DFN is a typical example for the hierarchic case: To provide its network connectivity services to scientific research organizations within Germany, DFN orders the required networking resources and services like optical fibers and managed wavelength services from 19 different sub-providers. An example for a heterarchic service is the public switched telephone network which is based on the combination of several networks operated by independent telephone companies. Another example is the IP service in the internet.

The management of service quality in the hierarchical case is based on Service Level Agreements (SLA) between the customer and the top-level provider, and contracts between the provider and its sub-providers. The hierarchical cooperation form is meanwhile well-understood. The service management frameworks ITIL and eTOM/NGOSS contain elaborated concepts for service quality management in provider hierarchies.

The heterarchic case is usually handled by a peering model using a best-effort strategy. The conditions for peering are agreed upon between two neighboring providers. This approach has significant drawbacks regarding service quality management: The overall service quality can not be assured if multiple domains are involved. An overprovisioning of resources is often applied to compensate this drawback.

However, these approaches are not always sufficient. For example, in the European Scientific Network Geant2 an end-to-end (E2E) link service is offered to support huge international research projects with exceedingly high bandwidth demand like the Large Hadron Collider at CERN. E2E Links are dedicated multi-gigabit Ethernet connections. Multiple independent national research networks are involved in the provisioning of E2E links. Despite of this heterarchic constellation, certain QoS parameters like service availability are demanded by the customers. Due to enormous costs for advanced optical

equipment, overprovisioning can not compensate the drawbacks of peering, so this approach can not be applied in this scenario.

As the classical solutions are not sufficient for heterarchic services with overall QoS assurance, new approaches have to be developed. Amongst other, the following questions have to be tackled:

Which roles are necessary to realize service quality management in such scenarios and how are they assigned to the involved actors?

How can the relationship between the customer and the provider be organized?

How can the cooperation between the providers be organized?

Which overall QoS parameters can be warranted?

How can local policies of the individual providers be taken into account?

How can central functions like monitoring, accounting and help desk be realized?

These issues are part of the research agenda of the EMANICS project. Elaborated solutions should be evaluated against real-life scenarios.

5.2.3.2 SLM and Service Provisioning Concepts

Service Level Management is tightly related to the provisioning of services, and especially the various types of services. A service is defined as a concept how a certain functionality is provided to a customer by a provider. Such a definition abstracts from the specific implementation while all customer-relevant service parameters (*e.g.*, Quality of Service (QoS) and cost parameters), such as agreed upon bandwidth or availability in network services, are specified in a Service Level Agreement (SLA).

The service provider is usually free to change the implementation of the service as long as the SLA is not breached. With such an approach, the growing number of resources needed to provide a service as well as the details about the processes needed for its implementation and operation are wrapped for the customer behind a more pleasant interface. This allows the service provider quite a degree of flexibility in the design and management of its resource infrastructure and enables it to leverage various economies of scale and scope, and has severe impact on the selected provisioning concepts.

In order to analyze service provisioning in more detail it is necessary to start with the service life-cycle. IT services can be associated with a service life-cycle that subsumes the steps from planning to termination of a particular service. A popular simple service life-cycle includes Plan-Build-Run. Service provisioning mainly deals with the plan, build, and change parts, providing the necessary input for run-time operations. It is therefore a major part of the service life-cycle. More precisely, service provisioning includes tasks such as planning new services, building the basic infrastructure, SLA negotiation and order processing, identifying adequate resources for service delivery or adapting existing services to specific customer's needs, specifying steps for service implementation and service operation, up to dynamic, near real-time service composition out of service modules based on customer requirements.

In order to offer services at competitive prices, a service provider must work efficiently by using as few resources as possible in terms of staff and machinery, while providing services effectively by meeting or exceeding the agreed SLA. Therefore, resources are typically shared between multiple customers as long as the SLA does not bind the service provider to use dedicated resources. With new virtualization technologies appearing on the market, the border between physically separated resources and shared resources becomes increasingly fuzzy.

Service level management and the agreed SLAs have influence on the allocation of resources and thus selected service provisioning concepts and mechanisms (*e.g.*, if the usage of dedicated resources is requested in a SLA). Service providers do not make any service guarantees when they provide best-effort services with a low complexity in service management and the usage of resources. Advanced quality of service mechanisms, *e.g.*, reserving a certain portion of end-to-end network bandwidth on demand with Resource Reservation Protocol (RSVP) for video traffic, need to be implemented in a way to fulfill the guaranteed service quality.

The selection of provisioning concepts depends also on the type of service. Services can be classified regarding a number of properties. For service provisioning, relevant aspects are mainly the number of customers that will subscribe to the service, the level of customization that these customers require, and the provided quality of service as agreed in SLAs. The number of service subscriptions directly translates to the number of times the order process will be executed, while the level of customization translates to the complexity of these processes and therefore the applicability to automated execution. The agreed quality of services imposes additional requirements on the provisioning process in terms of the allocation of either dedicated resources, high availability requirements which result in allocation of redundant resources.

Regarding the level of customization of service instances, IT services can be roughly grouped into (i) commodity, (ii) customized, and (iii) individual services. Commodity services are affordable, widely available, well-specified standard services with known functionality provided by many competing service providers. Examples include voice services, broadband internet access, or web hosting. The introduction of new commodity services is typically initiated by service providers sensing market demand and not by specific customer requests. They are intended to be subscribed by a large number of customers with a low level of customization. Customers have no choice in negotiating service parameters—the services are typically comparable even across service providers, mainly for interoperability. Prices are set by the service provider for pre-packaged service bundles with some service options and are hardly subject to negotiations. Therefore, customers usually have to accept the conditions stated by the service provider and can only search for a service offer that best fits their needs in terms of service features and price. With commodity services, the implementation of a service is typically hidden from the customer, with the service provider being free to change service implementation without prior notice. The high level of automation makes it possible to achieve a time range of seconds to days for completing the service provisioning process.

Individual services are sold to a small number of customers and individual negotiation of service parameters and values for service level objectives is an appropriate way to find a compromise in costs and customer requirements. Individual services have a very low number of orders, down to a single order by a major enterprise customer, typically organized as a long-term project. This is caused by the substantial size of a specific set of requirements expressed by the customer, which in turn requires a high level of customization.

Individual services start with the customer compiling a service request with detailed specifications of the customer requirements, usually expressed in terms at business process or service level. Many customers use consulting services in compiling this document. Multiple service providers analyze the service request and refine customer requirements and SLA individually together with the customer. Each provider identifies a set of appropriate resources and subservices to compose or build the solution. After estimating

costs and adding a profit margin, bids are sent to the customer. At this point, in most cases there is still a considerable level of uncertainty due to changing or non-specified customer requirements, estimates, and safety margins on both sides.

When the customer decides to accept an offer and places a service order, both sides agree on service functionality, service levels and costs. However, customers may demand specific interfaces, extensions or capabilities not commonly offered by suppliers, high availability requirements. In addition, often existing resources or services of the customer must be integrated with the new ordered service. Automation in provisioning of such requests is therefore in most cases not possible.

Provisioning is more project-driven, which involves many people communicating and negotiating with each other. Tools that support the collaboration of people such as project management tools, a knowledge base for past solutions and group communication tools provide valuable help and guidance.

Provisioning time is in the range of months to years. Key issues are the fast and thorough analysis of customer requirements and mapping them to needed resources, as well as good communication skills. Building individual services from flexible standard modules and experience in the composition of services can substantially reduce new developments and therefore provisioning time.

Customized services fill the wide range between the former two types of services. Service providers are often called “system integrators". In many cases, customized services are developed by adding more configuration and service options to existing commodity services to better meet the anticipated requirements of a potential set of customers.

Other providers develop customized services out from individual services. By refactoring the composition of the service into service modules and separating configuration options from service implementation, the former individual service becomes more flexible so that the requirements of more than one customer can be met and development costs per customer decrease.

Customized services typically have a substantial number of orders, but due to the number of service options and choices for the customer, hardly any two service orders are exactly the same. Depending on the specific service, customized services show features of both commodity and individual services. Customized services are typically built from standard service modules, which allows semi-automatic provisioning. A certain degree of flexibility for the customer is made possible by having the choice between different types of modules and a custom arrangement of the modules to a customized service. This results in a substantial growth of feature combinations with numerous dependencies. Automation would require to model all this complexity, which in turn requires models for all resources and subservices, and thus limits flexibility. The most appropriate level of automation is therefore determined on a case-by-case basis based on many criteria such as agreed service levels, cost of human labor, cost and flexibility of automation tools, uniformity of service orders, existing knowledge, supplier relationships and negotiation skills to name just a few.

6 Service Provisioning Concepts

Changing markets lead to service provisioning over administrative domains. This requires cooperation between multiple players and the definition of binding and measurable quality aspects. Key factors are efficiency of development, operation, and management of services

as well as the relation between quality level and cost. An increasing number of new high performance applications demand for network services with deterministic behavior.

Efficient and error-free service provisioning is critical to a service provider's business success. Provisioning systems and mechanisms that automate and streamline the provisioning process play thus an important role. The ability to deploy a service rapidly and to do so efficiently to keep costs down is crucial.

Dynamic service provisioning is certainly the aim and one of the most fundamental challenges a service provider faces. It addresses the vision that a customer is able to review a set of services, to select an appropriate service, to configure quality parameters, whereupon the service provider clicks on the "execute" button and the services are provisioned, monitored, and managed. In other words, when speaking of dynamic service provisioning, the automation of the different phases of the service provisioning process is addressed primarily.

In case of dynamic service provisioning a customer would select during the service ordering phase a service or a service package from available services that a provider offers to his customers. If a customer selects a service, he selects with this service functionality, service quality, and service costs, according to agreed values and thresholds in SLAs.

After the submission of the service order, the service provider would check the order with respect to contractual and technical issues and either rejects or grants the service order. Additionally, it is possible that a provider decides to negotiate the requested values. If a provider grants the service order, he starts the service provisioning and management of the service. Thus, the goal of dynamic service provisioning is to automate the provisioning and management steps. It depends on the service and the provider environment whether these steps differ very much. Case-based reasoning is an approach that could be used to apply similar steps (provisioning or management) to a dedicated provisioning situation.

Similar to the service provisioning within a provider environment, service provisioning among several domains takes has to take place. For example, if a user wants to have a video conference service within a certain time frame and dedicated points, this could result in service orders that had to be propagated throughout the service provider chains. If the service order has to be provisioned in certain hard time constraints, *e.g.*, has to be fulfilled in an hour, first the process has to be automated and second there should be no media break, *e.g.*, handling with paper.

6.1 Frameworks for Service Provisioning

Similar to Section 5.1, a brief outline of the concept of increasing relevance for IT service provisioning addresses organizational IT Service Management, represented by the IT business process frameworks ITIL and eTOM and their service provisioning approach [6].

6.1.1 Enhanced Telecom Operations Map (eTOM)

The eTOM is the NGOSS business process framework for Internet and Communications Service Providers (ICSP), for more details on NGOSS and its guidance on SLM, see Section 5.1.2.2. Its predecessor, the Telecom Operations Map (TOM) was first published by the TMF in 1998 and was superseded by the eTOM in 2001. The goal of TOM was the creation of an industry-owned framework of business processes, including the definition of a common enterprise-independent terminology for service management. Since 2004, eTOM is also an ITU-T Recommendation (M.3050).

The eTOM features different views on its processes, defining five view levels, from level 0 to level 4. Numerically higher levels offer an increasingly finer view of the service provider processes. In level 0, only three fundamental processes (in level 0 also called “process areas”) are defined, while on level 2, the only level for which process definitions for all level 0 process areas exist, there are already 72 processes. A further refinement into level 3 processes exists currently (eTOM Version 4.0) only for the Operations process area, and no level 4 processes have been defined yet.

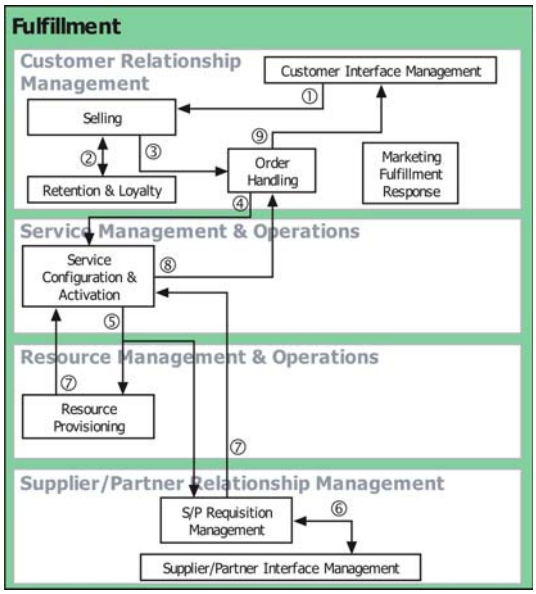


Figure 8: Simplified Fulfillment Workflow

While not being explicitly called „Service Provisioning“, it is the end-to-end process *Fulfillment* in the business area *Operations* that is responsible for providing customers with a requested service in a timely and correct manner. Figure 8 shows an exemplary service provisioning workflow between level 2 process elements:

1 *Customer Interface Management* routes an order for a new service instance to *Selling*.

2 *Selling* requests and receives a priority for the order from *Retention & Loyalty* based on customer data.

3 Prioritized order is handed over to *Order Handling*.

4 *Order Handling* passes a service activation request to *Service Configuration & Activation*.

5 *Service Configuration & Activation* decides which resources are to be supplied internally and issues resource reservation requests, work orders and resource activation requests to *Resource Provisioning* - simultaneously procurement requests are issued to *S/P Requisition Management* for externally supplied resources.

6 *S/P Requisition Management* chooses external sub-providers and issues orders through *S/P Interface Management* and waits for confirmation of external resource activation.

7 *Resource Provisioning* and *S/P Requisition Management* inform *Service Configuration & Activation* when all requested resources have been activated.

8 After final tests, *Service Configuration & Activation* informs *Order Handling* about successful activation of the service.

9 *Order Handling* instructs *Customer Interface Management* to inform the customer about completion of the order

Of course real service provisioning workflows within eTOM can be much more complex. The above example involves no pre-sales negotiations, no service design activities, nor does it show the necessary transfer of information to initiate *Assurance and Billing* of the new process instance. Also, like the workflow examples provided by the TMF in the eTOM addendum F (GB921F), the above description is limited to level 2 processes.

The decomposition of S*ervice Configuration & Activation* alone produces six level 3 process elements, all those would take part in the execution of *Service Configuration & Activation* in the above example. A level 3 workflow description of the above service provisioning example would therefore involve well above 20 process elements.

To summarize the analysis, the eTOM defines all generic process elements necessary for service provisioning, for each of which however it describes only the „what“, not the „how“. Thus, it does not provide ready-to-use internal or cross organizational workflows or interfaces. But the eTOM can lend a common structure to organization-specific workflows, significantly facilitate communications with customers and suppliers, and provide a basis for process automation efforts.

6.1.2 IT Infrastructure Library (ITIL)

Despite the similarity of their stated goals, the ITIL, stemming from a data center and mainframe computer background, and the eTOM, having its roots in a telecommunications services environment, follow quite different approaches. For a short introduction to the ITIL, see Section 5.1.2.1. There is no explicit „Service Provisioning“ process or another process correspondent to the eTOM Fulfillment process grouping defined in the ITIL. Where eTOM focuses on commodity or customized services offered to a multitude of customers, the ITIL concentrates on complex individual services offered to only a few customers (but possibly many users), cf. Section 5.2.3.2.

Consequently, eTOM differentiates clearly between the management of services (as in service class or product), addressed in the *Strategy, Infrastructure & Product* process area and the management of service instances, addressed in the *Operations* process area - a distinction that is not made with this precision in the ITIL. As a result, ITIL processes are not concerned with the efficient, automated provisioning of multiple service instances. Where eTOM's focus is on structuring and automating management activities for more or less standardized services, ITIL's focus is on coordinating the collaboration of IT personnel on not completely pre-determinable tasks and it puts most weight on support and coordination of operational activities and not the activities themselves.

In summary, ITIL gives primarily guidance on processes for planning, supporting and controlling operational activities and infrastructure changes. As such, practically all ITIL-ITSM processes have impact on efficiency of service provisioning. For further discussion on three ITIL processes of special relevance to service provisioning, namely S*ervice Level Management*, *Capacity Management*, and *Change Management*, refer to [6].

6.2 Network QoS Provisioning

Current IP networks provide only one simple service; that is the best-effort service. Under this service model, the network will try its best to deliver information packets as soon as possible to its destination. There is no guarantee though regarding the timeliness or the actual delivery of a packet. The absence of any kind of performance requirements has made it possible to run IP over a wide variety of link technologies, allowing routers to be

stateless, and has allowed the Internet to scale well in both the size of the network and the nature of the applications.

However, since the trend is for Internet to evolve into a global communication infrastructure, there is a need to provide more sophisticated service models in order to support emerging services and market demands, such as Voice over IP (VoIP), and Video Conferencing. Such services have Quality of Service (QoS) requirements that the current best-effort Internet can only provide with massive bandwidth overprovisioning. For that reason, IETF proposed the Integrated Services and Differentiated Services architectures for network QoS. These have been described in brief in Section 3.1 of the current Deliverable.

6.2.1 Mechanisms for Network QoS and Service Provisioning

Architectures for QoS provisioning alone, are not capable of providing QoS since if the amount and distribution/routing of traffic injected is not controlled, congestion will occur.

Therefore, on top of these architectures, mechanisms to control the amount and distribution of traffic are needed. These mechanisms are namely network provisioning for supporting network QoS and service admission control (subscription and invocation) for enforcing service performance objectives for individual service requests.

6.2.1.1 Network-based Service Provisioning

Network Provisioning is responsible for mapping the anticipated traffic onto the physical network resources. Traffic Forecasting techniques provide the anticipated demand, and Network Provisioning is responsible for determining the cost-effective (depending on the cost function used) dimensioning of the physical network resources into QoS Classes, subject to resource restrictions, load trends, SLS requirements, and policy directives and constraints [49].

Network Provisioning provides the directives in order to accommodate the forecasted traffic demands, and consequently, the anticipated SLSes. Network Provisioning essentially dimensions the network in terms of QoS Classes and routes. The primary objective of such provisioning is to ensure that the requirements of each subscribed SLS are met.

The design objectives can be further refined to incorporate other requirements such as: a) avoid overloading parts of the network while other parts are under loaded, and b) provide overall low network load (cost). One can formulate the provisioning problem as an optimization problem and solve it by using either optimization techniques or heuristic algorithms [49].

The output of Network Provisioning (routes and resources available for QoS classes) is further used for service admission control decisions.

6.2.2 Admission Control for Network-based Services

Service admission control can be further distinguished as taking place at two distinct phases: a) at service subscription, and b) at service invocation

6.2.2.1 Service Subscription

Service Subscription refers to the period during which a customer requests, negotiates, and agrees a service. At this phase static admission control can be performed, in the sense that the provider knows whether the requested long-term SLSes can be supported or not in the network given current configuration of network. This configuration is not an instantaneous snapshot of spare capacity, but the longer-term provisioning of the network [49].

Admission control at this level is necessary, in order to minimize the likelihood of overwhelming the network with customer contracts beyond some maximum capacity. The contract subscription constrains the customer's future usage pattern but at the same time guarantees a certain level of performance for invocations conforming to the agreement.

This is of benefit to the provider, who can use the information of the contract for network provisioning and traffic engineering purposes. It is also of benefit to the customer, as it provides a guarantee that network resources will be available whenever required. The subscription logic utilizes information related to the resource availability of the engineered network, to accommodate QoS services. Based on this information and related policy parameters, the subscription logic determines for a requested SLS, whether to accept it as is, to propose alternatives or to reject it.

The objective of the subscription logic is to determine whether service subscription requests can be accepted, with the purpose not to eventually overwhelm the network, while maximizing the number of subscribers.

6.2.2.2 Service Invocation

Service Invocation includes the process of dynamically dealing with a service request. It includes dynamic admission control on a per flow basis. Service Subscription provides to Service Invocation, the information of the service contracts. This information includes the SLA authentication attributes, as well as the technical SLSes.

Admission control at the Service Invocation level utilizes a view of the current available resources and when a request is admitted, or refused, Service Invocation delegates the

necessary rules to the appropriate Traffic Conditioners. When these rules are enforced, they ensure that packets are properly marked so that in- and out- of profile packets are handled appropriately.

The Service Invocation logic encompasses two different aspects. It manages the number and the type of the active services and it controls the volume of the traffic injected by the active services. The objectives of the invocation logic are to maximize the number of the admitted services while ensuring the QoS they enjoy matches the agreed in the relevant SLA, thus maximizing network utilization, while preventing QoS degradation caused by overwhelming the network with traffic it cannot properly handle.

Contrary to subscription logic, invocation logic relies on detailed network load measurements. Based on this information and related policy parameters, the invocation logic performs admission control through suitable algorithms so as not to overwhelm the network [49].

6.3 Provisioning Policies

Policies can be used to extend and guide the functionality of QoS management systems. This section—with reference to Section 6.2—will provide examples of how policies can influence network provisioning and service admission control.

Network provisioning policies can influence the behavior of the provisioning algorithm, setting constraints and priorities regarding the objectives that the administrator wants to achieve. Service admission control policies can be used to determine how conservative service subscription and invocation admission control can be, and what type of remedial actions can be taken to prevent congestion. The following sections provide more specific examples of how policies can influence network provisioning and service admission control.

6.3.1 Network Provisioning Policies

Network provisioning is an off-line resource management process. For such a static, off-line resource management process, two categories of policies can be identified [19].

The first category, initialization policies, concerns policies that result in providing initial values to variables used in the provisioning algorithm. These variables are essential for the functionality of the algorithm and do not depend on the state of the network but just reflect decisions of the policy administrator. The second category, resource provisioning policies, concerns those policies that influence the way the provisioning algorithm calculates the capacity allocation and the path creation configuration of the network. Examples of policies influencing network provisioning are as follows:

A parameter that can be defined by policies is the cost function used by the network provisioning algorithm. Using policies the administrator is able to either choose between a number of pre-specified cost functions and/or setting values to parameters in the desired function.

Another constraint that policies may add to a network provisioning algorithm is that of the maximum allowed number of hops. The administrator should have the flexibility to specify the maximum number of hops that routes are permitted to have. This number may vary depending on the QoS class.

Another constraint imposed by policies is the maximum number of alternative paths that the provisioning algorithm defines for the purpose of load balancing.

6.3.2 Service Admission Control Policies

Policies can also be used at both service subscription and service invocation level.

6.3.2.1 Service Subscription Policies

Service subscription policies can influence how conservative the subscription logic can be when deriving the admission control decisions. For example, the point up to which the administrator will allow subscriptions can be defined by policies, reflecting the provider's business objectives. The trade-off here is that the more subscriptions you accept (thus increasing your profit), the lower the guarantees you offer to your customers. Consequently, the behavior of the admission logic algorithm depends entirely whether the provider will target many and not so demanding customers or fewer with higher demands. Moreover, through policies, an operator might decide to reject a service subscription request despite that fact that there is spare capacity in the network to accommodate its traffic due to its business relations with that specific customer.

Another category of policies that influence the behavior of the Service Subscription is the negotiation logic policies. These policies are triggered every time the admission logic rejects a requested subscription. The purpose is to negotiate with the subscriber various alternatives until an agreement can be reached. Different alternatives can be proposed depending on the SLS characteristics, either in "all in one" mode or in "one by one" mode. The policies the administrator specifies determine the behavior of the negotiation logic, giving the flexibility to define different alternatives.

6.3.2.2 Service Invocation Policies

Service invocation policies can influence the behavior of the service invocation logic and the way it reacts and derives decisions in order to prevent congestion (meet QoS requirements of currently invoked services) and maximize network utilization.

Example of policy actions could be:

a) Adjust Service Rate,
b) Adjust Quality, and
c) Adjust Admission Control [19].

Adjust Service Rate is the equivalent of traffic policing. The configuration parameters to determine the profile the traffic will be policed against are expressed in terms of almost satisfied service rates and fully satisfied service rates. Fully satisfied service rate denotes the traffic rate, which if offered to an SLS, it enjoys QoS at its contractual rate while almost satisfied service rate denotes the rate which if offered to an SLS it enjoys QoS at a rate lower than its contractual rate but above what is considered acceptable for the particular type of service. The values of the above parameters are left to be decided by the administrator depending on the type of SLS according to business objectives.

Adjust Quality corresponds to packet remarking. Adjust Quality actions can be usually performed to qualitative services, downgrading or upgrading the quality they receive in terms of the QoS Class they are serviced by.

Adjust Admission Control corresponds to changing the parameters of the admission control algorithm to react to changes in the network load and become more or less conservative, as required by the specific condition.

6.4 Service Provisioning in Practice

Service provisioning in practice is done according to certain patterns that can be identified. A common approach in practice is to have overprovisioning. When buying resources or subscribing to subservices, and in-depth performance evaluation is often hard and costly to be performed for the service provider mainly due to uncertainties in future resource utilizations, subscriber behavior or introduction of new services. Thus, to be able to plan precisely according to history and expected behavior is quite difficult. On the other hand, overprovisioning of resources is typically cheap when introducing new hardware compared to cost of labor. Overprovisioning is thus a way to keep enough resources for changing or unpredictable demands. In addition, overprovisioning reduces the frequency of needed changes. As changes are one of the main reasons of service failures or disruptions, overprovisioning reduces potential downtimes.

One of the main reasons for using overprovisioning so widely around the service providers is simplicity in assigning services to run on specific resources. One of the main challenges of service provisioning is to find the optimal set of resources for a specific set of services (to be provisioning upon the resources). Since this is quite hard, overprovisioning is chosen which greatly simplifies the provisioning process at the cost of increased resource consumption and lower resource utilization.

Oversubscription is another quite common approach within service providers. Since usually customers do not make use of the full share of resources that a peak usage of the subscribed services would require, resources are underutilized. Thus, when planning the process for peak service usage, resources will be underutilized. If there are no specific customer requirements for (physical) dedicated resources, it is possible to increase resource utilization and the profit by oversubscription. Statistical models as well as continuous monitoring of the actual resource utilization are needed in order to avoid SLA beaches.

Virtualization techniques are also quite common in service provider environments and have also an impact on provisioning. They provide a view of dedicated resources in a shared-

resource environment to the users of the virtual resources. They considerable reduce the amount of complexity for the users by putting it into virtualization frameworks. Furthermore, they can reduce operational complexity by enabling functions that are hard to impossible to achieve on physical resources.

It should be noted that the level of detail at which service provisioning is done within provider environments differs quite a lot. The service provisioning process can be detailed to fill walls or just very rudimentary by focusing on overprovisioning. With respect to the processes, service provisioning is often done with respect to ITIL processes and employees are responsible for the ITIL manager roles. Nowadays, some customers explicitly demand service providers to be organized to ITIL standards.

Service provisioning approaches differ also if having a mass service to be provisioning (*e.g.*, thousand branch offices of a customer). In such a case it is necessary to have a rollout system (by consisting of a commercial database combined with a script-based workflow systems).

In general, relatively few service providers disclose information about their principles in provisioning, like on technical matters such as resource selection strategies, resource performance comparison or organizational matters such as processes or organizational models. It can be however summarized that due to the complexity of the service provisioning process, the data and tools required, a quite common approaches around services providers are overprovisioning and oversubscription.

7 Summary and Conclusions

Besides suitable technology and mechanisms, the provisioning of network services requires a well-defined set of economic measures and means in order to become economically viable. Only an economic management of network services that addresses technical and economic dimensions in an integrated manner is able to provide appropriate benefits for providers and to meet user needs at the same time.

While the previous deliverable D8.1 of EMANICS' WP8 covered mainly four areas, (a) the state-of-the-art in advanced economic network management, (b) the definition of key terminology, (c) the development of basic network and service management models, as well as (d) a discussion of important economic and technical mechanisms in support of an economic management of network services, deliverable D8.2 made one step forward by considering economic management dimensions going beyond the current state-of-the-art.

As the main purposes of deliverable D8.2 are to address the multi-provider dimensions of network and service management, as well as to cover service level management and provisioning concepts for network services, this deliverable has given an overview on all aspects of multi-provider environments. This included the full range of considerations on those roles adopted by involved stakeholders down to network QoS and service provisioning aspects. In particular, the use and application of SLAs as a highly important means for inter-domain contracts were covered. From an SLM point of view, those logical steps and entities involved in the introduction, negotiation and deployment of generic services have been given. Moreover, service provisioning concepts have been discussed, covering available frameworks, provisioning policies, and selected practical aspects. The common related work section gave a detailed overview on further work in these areas, additionally covering architectures for network QoS, multi-domain work, as well as SLM and service provisioning work.

The analysis of SLM models and mechanisms provides an excellent starting point for future research projects like task automation, where the development of solutions for a (partial) automation of SLM activities can be addressed. Additionally, integration supports concept development for interweaving SLM with other management disciplines, such as Accounting, Performance Management, and Configuration Management. Finally, the tasks of evaluation and verification will lead to the development of a methodology in helping the assessment and verification of SLM implementations with respect to framework recommendations and business objectives.

In general, formal modelling of service level agreements is in an early stage of development. Some promising approaches to a problem formalization have been devised, but so far these modelling approaches have not been tested in real scenarios. This constitutes an important field for future work.

Finally and in continuation of successful cooperations established between WP8 partners at the beginning of this project, which resulted already in a set of joint affiliation papers as reported in the last deliverable, these collaborations have again culminated in a set of papers as outlined in the selected cooperation work below, covering important details of the work that has been sketched in those sections above.

Based on this common ground, it will be important to critically assess and evaluate identified models, concepts, and mechanisms. Apart from prototypical implementations and experimental tests showing functional validity and performance of the selected approaches, generality determines an aspect of central importance for the presented work: For example, will an approach such as FCAPS still be valid in 5 years from now? Or will new technologies like multi-core processors fundamentally change the way network management will be done? What will be the impact of very high-speed networking technology beyond 10 GBit/s on service provisioning concepts? Accordingly, such questions will be addressed in one of the following steps.

8 Glossary

This section outlines again the major terms, which form the basis of those key multi-provider, and SLM models as well as of those service provisioning concepts described in this document. Additionally, it has been extended by those terms of relevant for D8.2.

Agreement
A common understanding about knowledge that is shared between two parties. Agreements are often assumed to be about policies or actions (agreements to act) and are often formalized using contracts in which case both parties agree to the terms of a common contract.

Architecture
An Architecture describes interactions of components of a complex system. Often, Architectures provide a layered or comparably structured view on the respective system.

Contract
a bilateral bundle of promises between two agents, that is intended to serve as the body of an agreement.

Framework
A Framework represents a reusable design for a system by describing concepts and structures that give guidance for the execution of a system's complex tasks without providing strict and mandatory implementation requirements or specifications. A Framework is often less concrete than a Model and described in a natural language rather than by using formal modeling techniques.

Model
A model is a representation or description designed to illustrate the structure or method of operation of an object, system or concept. In this capability, models are often used to simplify, down-scale and/or abstract from real-world entities.

Multi-domain
Adjective designating the characteristic (*e.g.*, of a management framework), that more than one administrative domain is involved. These domains often have to establish cooperation agreements on a peer to peer basis, coordinating aspects like configuration management or interaction strategies.

Multi-provider
Adjective designating the characteristic (*e.g.,* of a management framework), that more than one provider is involved. Multi-provider scenarios usually entail technical, operational and economic or legal cooperation issues between the participants to be solved by agreements.

Policies
A policy defines a course or method of action selected among alternatives and in light of given conditions to guide and determine present and future decisions.

Promise Agreement
A promise agreement is a pair of promises between two parties to acknowledge the content of an contract body.

Service
A service is the entity or unit of work offered by a service provider on behalf of a service consumer who can use the service. In general, a service includes several types of resources, including hardware- and software resources such as computing power, network links, storage capacity, and content, and it may even be composed of several sub-services.

Service Catalog
A Service Catalogue contains definitions of standard services as well as documentations of customer-specific services. It can be used as a foundation for automated service subscription or for the negotiation of SLAs.

Service Level Agreement (SLA)
A Service Level Agreement (SLA) defines the terms under which a service is offered to a service customer at a specific Service Access Point (SAP). The SLA includes a set of parameters which specify the service and the QoS under which it is provided (*e.g.*, the amount of bandwidth allocated, the involved session partners, metrics and algorithms that are used to compute SLA parameters), accountable units and the tariff which is used to charge for the service usage. Besides, several other aspects such as penalties or actions, respectively, to be taken if SLA objectives (*i.e.,* guarantees) are violated, trust relationships are part of a SLA.

Service Provisioning
IT services can be associated with a service life cycle that subsumes the steps from planning to termination of a particular service. A popular simple service life cycle is

called Plan-Build-Run, typically rerun after a Change or Improvement step. Service Provisioning mainly deals with the plan, build and change parts, providing the necessary input for run time operations. It is therefore a major part of the service life cycle. More precisely, service provisioning includes tasks such as planning new services, building the basic infrastructure, SLA negotiation and order processing, identifying adequate resources for service delivery or adapting existing services to specific customer needs, specifying steps for service implementation and service operation, up to dynamic, near real-time service composition out of service modules based on customer requirements.

9 References

[1] Agave Consortium: *Home Page;* http://www.ist-agave.org, June 2007.

[2] Ambient Networks Consortium: *Connecting Ambient Networks – Architecture and Protocol Design;* Deliverable 3.2, http://www.ambient-networks.org, March 2005.

[3] H. Asgari et al: *D01 ENTHRONE II Deliverable "Overall System Architecture – Version 2";* http://www-itec.uni-klu.ac.at/~timse/enthrone, February 2007.

[4] F. M. Bator: *The Anatomy of Market Failure*; Quarterly Journal of Economics, Vol. 72, No. 3, pp 351-379, August 1958.

[5] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss: *An Architecture for Differentiated Services;* December 1998, Internet RFC 2475.

[6] M. Brenner, G. Dreo Rodosek, A. Hanemann, H.-G. Hegering, R. König: *Service Provisioning: Challenges, Process Alignment and Tool Support;* in: Handbook of Network and System Administration, Elsevier, 2007.

[7] E. Brousseau, J.-M. Glachant: *The Economics of Contracts Theory and Applications*; Cambridge University Press, 2002.

[8] M. Burgess: *Analytical Network and System Administration—Managing Human-Computer Systems*; J. Wiley & Sons, Chichester, pp. 1-382, 2004.

[9] M. Burgess: *An Approach to Understanding Policy Based on Autonomy and Voluntary Cooperation*; IFIP/IEEE 16th International Workshop on Distributed Systems Operations and Management (DSOM 2005), pp. 97-108, 2005.

[10] J. Chen, A. McAuley, V. Sarangan, S. Baba, Y. Ohba: *Dynamic Service Negotiation Protocol (DSNP) and Wireless DiffServ;* International Conference on Communications (ICC 2002), New York, April 2002.

[11] A. Clemm, F. Shen, V. Lee: *Generic Provisioning of heterogeneous services—a close encounter with service profiles;* Computer Networks 43(1), pp. 43-57, 2003.

[12] B. F. Chellas: *Modal Logic: An Introduction*; Cambridge University Press, pp. 1-316, 1980.

[13] R. H. Coase: *The Problem of Social Cost*; Journal of Law and Economics, Vol. 3, No. 1, pp. 1-44, 1960.

[14] DAMMO Consortium: *Home Page;* http://www.csg.uzh.ch/research/dammo2/, June 2007.

[15] R. David, H. Alla: *Petri Nets for Modelling of Dynamic Systems—A Survey*; Automatica, Vol. 30(2), pp. 175-202, 1994.

[16] G. Dreo Rodosek: *A Framework for IT Service Management;* Habilitation Thesis, Ludwig-Maximilians-Universität München, 2002.

[17] Enthrone Consortium: *Home Page; http://*www.enthrone.org, June 2007.

[18] K. Farkas, S. Iyer, V. Machiraju, J. Pruyne, A. Sahai: *Automated Provisioning of Shared Services;* 10th Symposium on Integrated Network Management (IM 2007), Munich, Germany, pp. 848-851, 2007.

[19] P. Flegkas: *Policy-based Quality of Service Management in IP Networks;* Ph.D. Thesis, University of Surrey, April 2005.

[20] M. Garschhammer, R. Hauck, H.-G. Hegering, B. Kempter, M. Langer, M. Nerb, I. Radisic, H. Roelle, H. Schmidt: *Towards Generic Service Management Concepts – A Service Model Based Approach,* 7th International IFIP/IEEE Symposium on Integrated Management (IM 2001), 719–732, IEEE Publishing, IFIP/IEEE, Seattle, Washington, USA, May, 2001.

[21] M. Garschhammer, R. Hauck, B. Kempter, I. Radisic, H. Roelle, H. Schmidt: *The MNM Service Model – Refined Views on Generic Service Management;* Journal of Communications and Networks, Vol. 3, No. 4, pp 297–306, December, 2001.

[22] J. Gerke, P. Reichl, B. Stiller: *Strategies for Service Composition in P2P Networks;* International Conference on E-business and Telecommunication Networks (IICETE 2005), Reading, UK, 2005.

[23] J. Gerke, B. Stiller: *Service Support for Peer-to-Peer Networks;* Networking and Electronic Commerce Research Conference (NAEC 2005), Riva del Garda, Italy, 2005.

[24] J. Gerke, B. Stiller: *VETO—Enabling a P2P-Based Market for Composed Services;* 30th IEEE Conference on Local Computer Networks (LCN 2005), Sydney, Australia, 2005.

[25] G. Hardin: *The Tragedy of the Commons*; Science, Vol. 162 (3859), pp 1243-1248, December 1968.

[26] A. Høyland, M. Rausand: *System Reliability Theory: Models and Statistical Methods*; J. Wiley & Sons, New York, 1994.

[27] Information Systems Audit and Control Association (ISACA): *COBIT 4.1 – Control Objectives for Information and related Technology;* 2007.

[28] A. Keller, H. Ludwig: *Defining and Monitoring Service Level Agreements for Dynamic e-Business;* IBM T. J. Watson Research Center, 2002.

[29] H. L. Lee, S. Nahmias: *Single Product, Single Location Models*; In Logistics of Production and Inventory, Handbooks in Operations Research and Management Science, Vol. 4, Elsevier, 1993.

[30] X. Li, Z. Mi, X. Meng: *ASPOSE: Agent-based Service Provisioning in Open Services Environment;* LNCS 2881, pp. 42-52, 2003.

[31] Mescal Consortium: *Home Page;* http://www.mescal.org, June 2007.

[32] T. Nguyen, N. Boukhatem, Y. G. Doudane, G. Pujolle: *COPS-SLS: A Service Level Negotiation Protocol for the Internet;* IEEE Communications Magazine, May 2002.

[33] V. Sarangan and J. Chen: *Comparative Study of Protocols for Dynamic Service Negotiation in the Next-Generation Internet;* IEEE Communications Magazine, Vol. 20, No. 3, pp 151-156, March 2006.

[34] Office of Government Commerce (OGC): *Best Practice for Service Support,* 2000

[35] Office of Government Commerce (OGC): *Best Practice for Service Delivery,* 2001

[36] Oslo University College Computing Repository: *Introduction to Promise Theory*; http://research.iu.hio.no/promises.php, Accessed: April 2007.

[37] A. C. Pigou: *The Economies of Welfare*; Transaction Publishers, New Brunswick (NJ), USA, (originally published by Macmillian and Co., Ltd., 1952), pp 1-900, 2002.

[38] F. Reh: *Your Guide to Management (Free Newsletter);* http://management.about.com/cs/generalmanagement//keyperfind, June 2007.

[39] M. Salle, C. Bartolini: *Management by Contract*; 9th IEEE/IFIP International Network Management and Operations Symposium (NOMS 2004), pp. 787-800, 2004.

[40] S. Shenker, R. Braden and D. Clark: *Integrated Services in the Internet Architecture: An Overview;* June 1994, Internet RFC 1633.

[41] A. Smith: *The Wealth of Nations*; Penguin Books Ltd., London, United Kingdom, (originally published 1776), pp 1-602, 1999.

[42] B. Stiller, D. Hausheer, G. Schaffrath (Edts.): *Definition of Service Provisioning Goals, Economic Impacts, and SLA Management Tasks;* EMANICS Deliverable 8.1, June 2006.

[43] M. Subramanian, L. Lewis: *QoS and bandwidth management in broadband cable access network;* Computer Networks 43(1), pp. 59-73, 2003.

[44] TeleManagement Forum (TMF): *New Generation Operations Systems and Software – document library;* available at http://www.ngoss.org, 2007

[45] Tele Management Forum (TMF): *NGOSS SLA Management Handbook Volume 1—Executive Overview,* Morristown, USA, 2005

[46] Tele Management Forum (TMF): *NGOSS SLA Management Handbook Volume 2—Concepts and Principles,* Morristown, USA, 2005

[47] Tele Management Forum (TMF): *NGOSS SLA Management Handbook Volume 3—Service and Technology Examples,* Morristown, USA, 2005

[48] Tequila Consortium: *SrNP: Service Negotiation Protocol;* http://www.ist-tequila.org/deliverables, October 2001.

[49] P. Trimintzios: *Traffic Engineering for Quality of Service Provisioning in IP Networks;* Ph.D. Thesis, University of Surrey, March 2004.

[50] N. Wang et al: *D1.1 AGAVE Deliverable "Parallel Internet's Framework";* http://www.ist-agave.org, September 2006.

[51] X. Wang, H. Schulzrinne: RNAP: a Resource Negotiation and Pricing Protocol; International Workshop on Network and Operating Systems Support for digital Audio and Video, June 1999.

[52] S. Van den Bosh, G. Karagiannis, A. McDonald: *NSLP for Quality of Service Signalling;* IETF Internet Draft, February 2005.

[53] J. von Neumann, O. Morgenstern: *Theory of Games and Economic Behavior*; Princeton University Press, Princeton, 1944.

10 Abbreviations

A4C	Authentication, Authorization, Accounting, Auditing, and Charging
AF	Assured Forwarding
AGAVE	A Lightweight Approach for Viable End-to-end IP based QoS Services
ASP	Application-level Service Provider
ASPOSE	Agent-based Service Provisioning in Open Services Environment
ATM	Asynchronous Transfer Mode

CBR	Case-Based Reasoning
CDC	Caisse des Dépôts et Consignations
CC	Content Consumer
CCP	Customer Contact Point
CCPR	Customer Contact Point Reference
CIM	Common Information Model
COBIT	Control Objectives for IT and related Technology
COPS	Common Open Policy Service
CP	Content Provider
CPA	Connectivity Provisioning Agreements
cSLA	Customer Service Level Agreement
CSM	Customer Service Management
DAMMO II	Distributed Accounting and Auditing Management for Multiple Mobile Network Operators
DiffServ	Differentiated Services
DS	DiffServ
DS	Deliver and Support
DSCP	Differentiated Service Code Point
DSNP	Dynamic Service Negotiation Protocol
DVB	Digital Video Broadcast
E2E	end-to-end
EF	Expedited Forwarding
EMANICS	Management of the Internet and Complex Services
ENTHRONE	End-to-End QoS through Integrated Management of Content, Networks and Terminals
eTOM	Enhanced Telecom Operations Map
EU	European Union
FCAPS	Fault, Configuration, Accounting, Policy, and Security Management Model
GSLP	Generic Signaling Layer Protocol
HIO	Oslo University College
IC	Imperial College
ICP	IP Connectivity Providers
ICSP	Internet and Communications Service Providers
IETF	Internet Engineering Task Force
IP	Internet Protocol
INRIA	Institut National de Recherche en Informatique et Automatique
ISACA	Information Systems Audit and Control Association
ISO	International Organization for Standardization
IT	Information Technology
ITIL	IT Infrastructure Library
ITSM	IT Service Management
KPI	Key Performance Indicator
KQI	Key Quality Indicator
MbC	Management by Contract
MESCAL	Management of End-to-End Quality of Service Across the Internet at Large
MPEG	Motion Pictures Expert Group
NGOSS	New Generation Operations Systems and Software

NP	Network Provider
NSLP	NSIS Signaling Layer Protocol
OGC	Office of Government Commerce
OLA	Operational Level Agreement
PCP	Physical Connectivity Provider
pSLA	Provider Service Level Agreement
QoS	Quality of Service
RNAP	Resource Negotiation and Pricing Protocol
RSVP	Resource Reservation Protocol
SAP	Service Access Point
SC	Service Client
SDF	Service Degradation Factor
SE	Service Element
SIP	Service Improvement Program
SLA	Service Level Agreement
SLM	Service Level Management
SLR	Service Level Requirement(s)
SLS	Service Level Specification
SP	Service Provider
SQP	Service Quality Plan
SrNP	Service Negotiation Protocol
TMF	TeleManagement Forum
TOM	Telecom Operations Map
ToS	Type of Service
UC	Underpinning Contract
UMA	Universal Media Access
UniBwM	University of Federal Armed Forces Munich (Universität der Bundeswehr)
UniS	University of Surrey
UniZH	University of Zürich
UPC	Universidat Politecnica de Catalunya
UPI	University of Pitesti
UT	University of Twente
WP	Work Package
WSLA	Web Service Level Agreement

11 Acknowledgements

This deliverable was made possible due to the large help of the WP8 team of the EMANICS team within the NoE, which includes besides the deliverable authors as indicated in the document control, Matthias Hamm (Leibniz Computing Center), Mark Yampolskiy (Leibniz Computing Center), Marinos Charalambides (UniS), George Pavlou (UniS), Ning Wang (UniS), Cristian Morariu (UniZH), and Peter Racz (UniZH). Many thanks to all of them.

12 Selected Cooperation Work

This section of selected cooperation work covers a range of work started recently in the context of EMANICS WP8 as well as sometimes shortly before. Thus, the content in this section consists out of paper abstracts and summaries from single institutions (early starts) as well as joint work between EMANICS partners (recent starts).

To provide an overview of these areas of work, the following subsections list authors, abstracts, and the table of content, if they exist, or a respective sketch of the idea to be worked on in the next month of EMANICS. In case of a full paper being available, it is part of this deliverable at its final end, following the sequence as their summaries below.

Thus, D8.2 covers 3 single affiliation papers and 1 joint affiliation papers being ready and 1 single as well as 2 joint affiliation papers being under preparation, totalling 7 papers.

12.1 A Protocol to Support Multi-domain Auditing of Internet-based Transport Services

Authors: Frank Eyermann, UniBwM, Burkhard Stiller, UniZH

Abstract:

Auditing of Service Level Agreements (SLA) defines the process of monitoring whether a service provider delivers agreed upon service levels or not. While frameworks exist to monitor application-level SLAs, the end-to-end monitoring of IP-carrying SLAs, especially in a multi-domain environment like the Internet, is still an open issue.

This work analyzes methodologies how IP-carrying SLA parameters could be efficiently measured and develops a protocol, which supports this process and minimizes overhead.

Table of Contents:

Summary and Outlook

Publication:

2nd International Conference on Internet Monitoring and Protection (ICIMP 2007), Silicon Valley, U.S.A., July 1—6, 2007. Organized by the International Academy, Research, and Industry Association (IARIA).

12.2 Frameworks for Business-driven Service Level Management

Authors: Thomas Schaaf, LMU

Abstract:

In the majority oft today's IT organizations, Service Level Agreements (SLA) are an important means for underpinning IT service provisioning by clearly defined Quality-of-Service (QoS) parameters as well as service costs and violation penalties. The management of those SLAs is the main subject addressed by the discipline of Service Level Management (SLM), which covers several activities vital for the deployment of customer-oriented, high-quality and well-performing IT services.

This paper analyzes and compares two of the most important SLM frameworks available in business-driven IT Management: the IT Infrastructure Library (ITIL) with its SLM reference process and the NGOSS SLA Management Handbook. In order to deliver significant and helpful results, we derive a set of evaluation criteria from a realistic IT scenario. These criteria are applied to ITIL and NGOSS in order to elaborate possible areas of conflict as well as complementary fields and unaddressed issues. The results are visualized in an analysis matrix, which shows whether and how ITIL and NGOSS may co-exist as SLM frameworks in one operative and management environment.

Publication:

2nd IEEE/IFIP International Workshop on Business-driven IT Management (BDIM 2007), Munich, Germany, May 21, 2007.

Acknowledgements:

The author wishes to thank Stylianos Georgoulas (University of Surrey, UniS) for helpful discussions and valuable comments on previous versions of this publication.

12.3 Accounting and Charging—Guarantees and Contracts

Authors: Burkhard Stiller, David Hausheer, Jan Gerke, Peter Racz, Cristian Morariu, Martin Waldburger, UniZH

Abstract:

Charging for IP-based communications determines the overall term for metering or monitoring, accounting, pricing, charge calculation, and billing. These five actions are detailed in this chapter to provide a clear view on their interdependencies as well as their relations to distributed computing. Since an ubiquitous computing approach does require communication means between all entities involved, the provisioning of these communication channels is supported typically by commercial service providers—covering network, transport, and value-added services. Thus, the legal and contractual relationships between customers and providers as well as technological choices of protocols, mechanisms, and parameters define the area of interest here.

Table of Contents:

Publication:

Book Chapter in “Ubiquitious Computing” Textbook, to be published in Ideas Group Publishing (IGI), New York, U.S.A., 2007.

12.4 Evaluation of an Accounting Model for Dynamic Virtual Organizations

Authors: Martin Waldburger, UniZH, Matthias Göhner, UniBwM, Helmut Reiser, LRZ München, Gabi Dreo Rodosek UniBwM, Burkhard Stiller, UniZH

Abstract:

With the ongoing trend of adopting Grid systems as a means for service-oriented computing in dynamic Virtual Organizations (VO), the need for appropriate support mechanisms becomes apparent. Accounting of Grid resource and service usage determines the central support activity as it prepares accounting records that provide the main input for analysis, optimization, and in particular for charging and billing purposes.

An embracing study of existing Grid accounting systems has revealed that these approaches focus primarily on technical precision and on project-specific issues. However, existing systems do not support multi-provider scenarios or virtualization concepts—both key requirements for service provisioning and the according accounting in dynamic Virtual Organizations. Moreover, existing Grid accounting approaches are not based on the appropriate economic accounting principles. Consequently, a resource-based, highly flexible accounting model for dynamic Virtual Organizations was developed, combining both, technical and economic accounting by means of Activity-based Costing (ABC) service constituent parts and defined accountable units.

Driven by a successful preliminary functional evaluation, this paper pursues a full-fledged evaluation of the developed Grid accounting model. This is done for the specific

environment of the Leibniz Supercomputing Centre (LRZ) in Munich, Germany. For that purpose, the detailed evaluation methodology as well as the evaluation environment is outlined, what leads to actual model-based cost calculations for a defined set of considered Grid services. Gained results are analyzed and the respective conclusions on model applicability and potential optimizations are drawn.

Table of Contents (Planned):

Introduction

Related Work

- DVO Service Models
- Grid Accounting Models for DVOs
- Grid Resource Classification

Evaluation Methodology

- Key Evaluation Objectives
- Accounting Model Application Methodology
- LRZ Scenario Definition

Results

Discussion

Summary and Conclusions

Publication:

Under preparation

12.5 Decentralized Auctions for Bandwidth Trading over Optical Links in a Virtual Network Environment

Authors: David Hausheer, UniZH, Aiko Pras, UT, Burkhard Stiller, UniZH

Abstract:

Recent technical advances in telecommunications, especially in the area of optical fibre technology, have led to a tremendous increase of bandwidth, based on which the support of Quality-of-Service and diverse Internet-based application services such as VoIP and IPTV become possible. However, suitable bandwidth trading mechanisms for such services, which require in many cases short-termed bandwidth assignments, e.g., for large sporting events or cultural open air activities, have not yet evolved. In particular, there is currently no technically and economically feasible solution in place, enabling service providers to buy the necessary bandwidth for such services “on demand” and resell it to other providers if not used. This paper presents how PeerMart, a fully decentralized auction-based marketplace based on peer-to-peer (P2P) networking principles, can be applied to trade bandwidth services in an efficient and scalable manner. A specific scenario is developed, targeted at trading bandwidth over optical links in a fully virtualized networking environment, from which detailed requirements are derived. The particular technical and economic aspects of the scenario are investigated and the specific service parameters defined. Finally, PeerMart’s basic design supporting generic services is refined to meet the specific requirements of the considered bandwidth trading infrastructure.

Table of Contents (Planned):

Introduction

Scenario and Requirements
- Virtual Network Model
- Virtual Node Model
- Bandwidth Trading Scenario

PeerMart

Bandwidth Trading Infrastructure
- Service Parameters
- Reselling Bandwidth Services

Related Work
- Architectures for Decentralized Service Markets
- Decentralized Auctions
- Optical Network Management Systems

Conclusions and Future Work

Publication:

Under preparation

12.6 Peering Agreements

Authors: Aiko Pras, Remco van de Meent, UT

Abstract:

Peering agreements, together with IP transit agreements, govern the exchange of Internet traffic between network operators. Depending on network architecture, size, commercial and other interests, operators work with a spectrum of agreements ranging from `full transit' to `transit free'. Full transit means that one pays another network for routes to all of the Internet . Contrarily, transit free means that a network operator carries its traffic to (nearly) all other network operators in the world by itself. Clearly, transit free operations require a rather dense global network; it is so far achieved by only a few large network operators.

In between full transit and transit free operations are partial transit and partial peering operations. In partial transit agreements, one pays the supplier for instance for access to only a regional part of the suppliers global network. Partial transit is typically used as a supplement to one or more full transit agreements, for instance to get better routes to a specific region. In a partial peering agreement, parties agree to exchange only regional traffic; this is for instance seen in the US where parties in the East Coast only exchange traffic originated/destined in the East Coast.

Network operators can have various peering policies. Some, commonly small network operators have a very `open peering policy': they like to peer with everybody who likes to peer with them. Peering relations can be laid down in a formal contract, but it is often seen that some informal e-mail exchanges suffice to setup a peering relation when both parties have an open peering policy. Another type of peering policy is the `restrictive peering policy', limiting the possible peering relations. For instance, a large operator may only want to peer (settlement free) with other large operators, and only offer (paid) IP transit agreements to smaller operators. Strategic and/or tactical concerns may influence the kind of peering policy operators have, and this policy can differ depending on the region: in a region where an operator is relatively big, it may have a much more restrictive peering policy than in a region where the same operators is relatively small.

Table of Contents:

Publication:

Under preparation

Acknowledgements:

Various network operators who choose to remain anonymous.

12.7 Admission Control for Inter-domain Real-time Traffic Originating from Differentiated Services Stub Domains

Authors: Stylianos Georgoulas, George Pavlou, Panos Trimintzios, Kin-Hon Ho (UniS)

Abstract: Differentiated Services (DiffServ) are seen as the technology to support Quality of Service (QoS) in IP networks in a scalable manner by allowing traffic aggregation within the engineered traffic classes. In DiffServ domains, admission control additionally needs to be employed in order to control the amount of traffic into the engineered traffic classes so as to prevent overloads that can lead to QoS violations. In this paper we present an admission control scheme for inter-domain real-time traffic originating from DiffServ stub domains; that is real-time traffic originating from end-users connected to a DiffServ stub domain towards destinations outside the geographical scope of that domain. By means of simulations we show that our scheme performs well and that it compares favorably against other schemes found in the literature.

Table of Contents:

Publication: 5th International Conference on Wired/Wireless Communications (WWIC 2007), Coimbra, Portugal, in: Lecture Notes in Computer Science, Springer, Heidelberg, Vol. 4517, May 23-25, 2007, pp. 115—128.

A Protocol to Support Multi-domain Auditing of Internet-based Transport Services

Frank Eyermann
Information Systems Laboratory (IIS)
Universität der Bundeswehr München
Munich, Germany
Frank.Eyermann@unibw.de

Burkhard Stiller
Communication Systems Group CSG, IFI
University of Zürich
Zürich, Switzerland
stiller@ifi.unizh.ch

Abstract—Auditing of Service Level Agreements (SLA) defines the process of monitoring whether a service provider delivers agreed upon service levels or not. While frameworks exist to monitor application-level SLAs, the end-to-end monitoring of IP-carrying SLAs, especially in a multi-domain environment like the Internet, is still an open issue.
This work analyzes methodologies how IP-carrying SLA parameters could be efficiently measured and develops a protocol, which supports this process and minimizes overhead.

Keywords: SLA; Auditing; SLA Monitoring; Performance Metric; End-to-End Measurement

I. INTRODUCTION

An adequate access to the Internet, mainly in terms of reliability, became an Achilles' heel for many businesses. If a web-shop is not available or reacts too slowly on requests, customers buy somewhere else. But also in Business-to-Business (B2B) communications [14] it is important carrying out transactions in time. A disruption of network connectivity or even only a significant degradation of the service quality could result in a financial loss. To reduce these risks, companies utilizing the Internet as a means for commercial communications contract Service Level Agreements (SLA) [4] with Internet Service Providers (ISP). SLAs manifest those service levels the ISP will provide and the customer needs to pay for. Most valuable for customers are SLAs guaranteeing network connectivity end-to-end [9], i.e. from an origin of a connection the whole path to the destination, even in multi-domain scenarios, when the destination can be reached only through another provider as the one the origin's access is based on.

SLAs as well as all other agreements need to be monitored: Does the contractual partner fulfill its obligations as stated? This task is referred to as SLA Monitoring or Auditing [12], [7], two terms which are used in this context synonymously. Auditing can be regarded in an abstract manner as the comparison of nominal with actual values. For IP-carrying SLAs, nominal values are provided by the SLA, actual values need to be measured in the network for a given situation. In order to measure unambiguously actual performance values, metrics need to be defined, which describe the measurement setup and the relevant process in sufficient detail.

In a multi-domain scenario, like the Internet, where different network operators need to cooperate in order to provide an end-to-end connection, measurement also needs to determine where, i.e. in which domain, SLA parameters have been violated. This increases the measurement effort drastically and, therefore, makes auditing of IP transport quite "expensive". This paper presents a solution to the problem of determining a promising approach to reduce the measurement effort in a multi-domain scenario.

Section II of this paper presents related work and introduces important terms for this work. In Section III different measurement techniques are analyzed and their suitability for auditing is evaluated. Section IV presents the design of a new protocol called MeSA (Measured Signaling for Auditing), which helps to minimize the measurement effort in multi-domain environments. While Section V evaluates the approach taken, finally Section VI summarizes the key achievements and draws conclusions.

II. RELATED WORK

Related work on the problem statement given above is basically addressing three areas: (1) the description with which agreements between contracted partners can be formalized, (2) measurement approaches, and (3) auditing systems. Based on an appropriate combination of these three aspects the basis for auditing multi-domain transport services is laid.

A. Service Level Agreements

SLAs are a well known concept to manifest services and the way how these services are delivered between a service provider and a service consumer [4]: This approach is valid in various business branches, starting from housekeeping and ending in Internet-based transport service deliveries.

In general, an SLA consists of three parts [12]: The *involved parties, SLA parameters,* including metrics and algorithms used to compute them and *Service Level Objectives (SLO) and actions,* which have to be taken upon violation of SLA parameter thresholds.

The more enterprises need to depend on the Internet, the more SLAs have been applied in IT businesses as well. Most SLAs also contain bonus and penalty regulations for the case, when a provider delivers services very well or when SLOs are violated. An SLO defines a single quantity, which can be measured individually and for which the SLA defines a threshold to be met by the provider. Thus, SLAs determine an

This work has been performed partially in the framework of EMANICS (FP6-2004-IST-4-026854-NoE).

approach to share the risk of financial loss, when agreed upon services are not delivered as contracted.

Unfortunately, in the area of network provisioning, which is fundamental for all IT branches, currently a low dissemination of SLAs can be observed. The reason for this is that nowadays the Internet lacks the possibility to measure accurately and adequately and to report on given SLAs, thus, making agreements themselves hard to be implemented [4].

B. IP Performance Metrics

The IP Performance Metrics working group (IPPM WG) is a working group of the Internet Engineering Task Force (IETF). Its goal is to define metrics to be applied to quality, performance, and reliability of Internet data delivery services [10]. Additionally, the working group defined a general framework for accurately measuring and documenting metrics [21]. Metrics in this sense are carefully specified quantities related to the performance and reliability of the Internet.

Metrics are essential whenever formal and practical definitions of performance parameters are needed: For example, for an SLA containing an SLO with an upper limit for delays, it is mandatory for each partner to have the same understanding of delay and its measurement.

The IPPM WG offers neither definitions nor suggestions on *how* performance parameters are measured. Their emphasis is on definitions and the unambiguous understanding *what* a parameter expresses in order to allow measurement results to be compared, shared, and validated by different entities.

Metrics defined by the IPPM working group can be used as building blocks to compose SLOs for an SLA. In contrast to the bottom-up approach of IPPM, Section III.A analyses measurements from a top-down perspective, looking which performance metrics on the network layer influence the performance of distributed applications.

C. Auditing

Auditing can be generally defined as the act of assessing the validity of a process according to rules [7]. A financial audit, *e.g.*, determines if financial statements of a company are in accordance with the law.

In IT the term auditing has two meanings. A security audit is the search through a computer system for security problems and vulnerabilities [25]. This is mostly performed by inspecting in an offline manner existing system logs for suspicious entries or known attack patterns. The second meaning is the one referred to in this work: auditing as the process of checking the compliance of actual provided service quality with an SLA specification.

Related work in this area can be found in the *Web Service Level Agreement* (WSLA) framework [12]. It is a system to measure and monitor QoS parameters for application-level SLAs, especially for Web Service. A language to express SLAs and a run-time architecture to remotely monitor web services is included. Even though the authors propose a very flexible system, it is specialized for application-level SLAs and, as the end-to-end character of network SLAs is not appropriately represented, less suitable for network SLAs.

An approach for auditing of network SLAs can be found in [27]. Auditing here is an addition to an Authentication, Authorization, and Accounting framework based on the AAA standard protocol "Diameter" [2]. Not further defined "QoS equipment" is part of the service equipment and performs the actual auditing. However, the auditing process itself, i.e. the measurement of single performance parameters, is not described, so a central question remains open. An open framework for Auditing has been developed in [7].

The Distributed Accounting and Auditing for Multiple Mobile Network Operators (DAMMO, [5]) project defines auditing procedures and data exchange in close detail. A4C-Servers (Authentication, Authorization, Accounting, Auditing and Charging-Server) perform in collaboration with service equipment the actual auditing. Auditing procedures for handovers and network changes are specified. Again, the measurement and validation process is not defined. It is assumed service equipment is able to perform these tasks.

III. Measuring of Performance

Once metrics and frameworks as well as mechanisms for an auditing purpose are defined, the analysis of existent possibilities on how to verify SLOs agreed upon in SLAs is essential. This happens by discussing in Subsection A, what typical SLOs in IP carrying SLAs could be and by explaining in Subsection B, how these could be measured in practice.

A. Performance Metrics

An SLO has been defined in Section II.A as a single quantity, which can be measured individually and for which a threshold exists. This definition of IPPM could also be applied to describe a performance metric with a threshold.

But which performance metrics can be found in SLAs and with which thresholds? The answer needs to be derived from traffic profiles of typical Internet applications. Application categories being of interest for auditing include:

- Real-time applications (*e.g.*, Voice-over-IP)
- Multimedia applications (*e.g.*, Video streaming)
- Bulk-data transfer (*e.g.*, GridFTP download)
- Client-Server applications (*e.g.*, distributed simulation)

Analyzing these applications the following transport layer performance parameters can be identified [6]:

- Throughput (i.e. available Bandwidth)
- Per-flow sequence preservation
- Packet loss
- One-way delay
- Round-trip time
- Jitter

Availability is no transport layer performance parameter. From a flow-level point of view a hardware failure is seen as complete packet loss over a longer interval. Still, availability is stated in many SLA as a time share (*e.g.*, 99.9%) at which the system is available to provide services. It needs to be monitored, even if no application is sending any data.

B. Measuring Performance Metrics

After having identified relevant performance parameters the next step is to analyze how they can be measured in order to monitor SLOs. Different methodologies to measure parameters exist, but none of which is universal in the sense that it is useful for each performance parameter. The most common measurement methodologies include:

- Passive measurements monitor only packets and evaluate information in packet headers and signaling information. For the measurement itself no additional network traffic is generated, but during the measurement huge amounts of data may accrue, which have to be transferred afterwards to another host for off-line evaluation and correlation.
- Brokers keep an account on vacant resources in the network and allocate them to services.
- Counters and gauges are maintained internally by routers and other network equipment in order to provide information on their status and the status of the network (*e.g.*, as part of Management Information Base, MIB).
- Probing is the injection of artificial packets (so called probes) in the network. Special measurement agents are placed in the network, which monitor how probes are treated by the network. Only probes are metered.
- Inference techniques, sometimes referred to as network tomography, are a subclass of probing techniques, for which measurement agents are only deployed at the source, mostly combined in a single program with the probe generator. This technique uses standard behavior of protocol stack implementations on nodes in the network to receive a feedback to requests the program sent. The most widely know tool using such a technique is traceroute, which sends probes with increasing values in the TTL-field of the IP header in order to achieve information about intermediate systems on the way to a destination host, from time-to-live exceeded ICMP-messages returned by routers.

C. Applicability of Measurement Methods to Performance Parameters

These measurement techniques presented in Section III.B have different strengths and weaknesses when used to measure the performance parameters listed in Section III.A. In order to reduce the measurement overhead and improve the measurement accuracy, it is beneficial to choose always the most appropriate measurement technique.

This "appropriateness" is shown in Table I. Within this table all different methods are rated in 6 categories:

- Well suited (++): measurement is possible with good accuracy and less effort.
- Suited (+): measurement is possible, but accuracy is lower and/or effort is higher than for well suited ones.
- Possible (0): measurement is possible, but accuracy is low and/or effort is significant.
- Difficult/very difficult (-/--): measurement is theoretically possible, but not economically achievable. The effort necessary is much higher than the gain, especially as there are better suited methods.
- Not possible (X): measurement is not possible, even with a high effort.

TABLE I. COMPARISON OF MEASUREMENT METHODOLOGIES FOR PERFORMANCE PARAMETERS

	Connectivity	RTT	Throughput	Sequence	Loss	Delay	Jitter
Passive	0	-	++	++	++	-	-
Broker	X	X	+	X	X	+	-
Counters Gauges	0	X	0	X	0	0	-
Probing	++	++	-	X	+	++	+
Inference	++	++	-	X	+[1]	+	-

++ = well suited, + = suited, 0 = possible,
- = difficult, -- = very difficult, X = not possible

In a more detailed discussion on TABLE I. another criteria has to be included. Each performance parameter has special requirements, where in the data path measurement has to be performed. Four different locations can be identified:

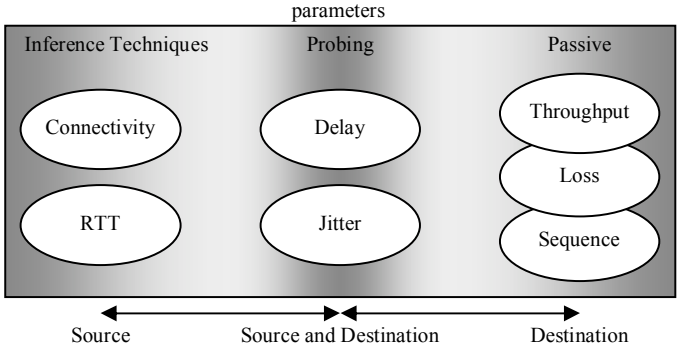
- Anywhere in the data path
- Only at the source
- Only at the destination
- At source and destination

Between efficient measurement technologies and measurement locations correlations can be found:

- A passive measurement is well suited if only a measurement at the destination is necessary (*e.g.*, loss and sequence). Information from the source needs to be communicated to the destination or a central measurement coordination station, which raises the effort drastically (*e.g.*, delay and jitter).
- Brokers show typically only a general overview on the network. They can only warn, when a state of the network is reached, where load-dependent performance parameters might degrade (*e.g.*, throughput and delay). When congestion in the network is detected, brokers can not measure the impact in detail.
- Counters and gauges can provide information on the state of routers. Quite similar to brokers, counters and gauges can warn, if a state is reached threading network performance. Thus, they allow only monitoring the networks health status, but can not provide detailed values for performance parameters.
- Probing has its strength, if the actual traffic does not need to be looked at, because the measurement is only performed with these probes. Probes can be created in an application-specific manner, which allows for measuring at different places in the network, i.e. at the source and the destination.
- Inference techniques are a subclass of probing techniques, and they show quite similar strengths and weaknesses. But since no measurement infrastructure is in place, probes have to be reflected to sources, which might cause additional distortion. Therefore, this technique is best suited for performance parameters, which can be measured at the source.

[1] Values might be quite inaccurate.

Figure 1. Relationship of measurement methodologies and performance parameters



In summary, Fig. 1 shows these relationships. Inference techniques deployed at the source are best suited for measuring connectivity and round-trip time, while probing with measurement agents at least on the destination can be used for measuring delay and jitter. Passive techniques are best deployed on the destination and can measure loss and per flow sequence preservation.

D. Auditing in Multi-domain Scenarios

Based on the discussion above, it can be seen that it is beneficial for auditing to use different measurement methods at different locations in the data path in order to work efficiently. This has considerable consequences for auditing in a scenario, where more than one ISP has to be traversed between a sender and a receiver. This is often referred to as the multi-domain characteristic [5].

Fig. 2 shows an example, where a sender is connected over three different networks to a receiver. It is assumed that each network represents one domain. On one hand, these ISPs need to work together in order to transfer data from a sender to a receiver, on the other hand, they determine independent business entities with their own goals and policies. Recent history has shown that ISPs are not willing to accept restrictions in the autonomy over their networks [13].

Multi-domain auditing now requires that each single domain fulfills its SLA, which has to be examined, too. This means, that within each domain all relevant performance parameters have to be metered. This is displayed in Fig. 2. Throughput, round-trip time, connectivity loss, and per-flow sequence preservation is measured (1) at the appropriate places for the whole scenario, i.e. source and destination which are sender and receiver respectively, and (2) within each domain, i.e. source is the ingress router. The destination is the egress router of the domain. The auditing overhead, therefore, increases drastically. In this scenario delay for examples has to be measured four times, i.e. within each domain and end-to-end. The MeSA protocol, developed and described in the following section, reduces this overhead to a minimum, but still complies with multi-domain requirements and ISP autonomy.

IV. MeSA Protocol Design

As seen, a significant effort has to be made performing auditing. Especially the multi-domain characteristic, requiring measurement within each domain, boosts complexity. A novel protocol termed MeSA (Measured Signaling for Auditing) defined here can reduce this effort by combining measurement and signaling. Before details of this protocol are presented in Subsection 0E and the following, major design decisions are discussed.

A. Measurement Scope

Most of today's professional routers are able to monitor path conditions (*e.g.*, Active Traffic Monitoring implemented in Cisco IOS, [26]). It is possible to program the devices to monitor loss and round-trip time between itself and a preconfigured destination device. The device will periodically send probes to the destination device, which returns these packets.

With the help of this feature an operator can monitor all links in the network and create statistics, which can be used to generate a detailed picture of major conditions in a network. If all paths in the network are monitored, the picture is complete and for any flow in the network it is theoretically possible to approximate the flow's performance parameters.

The key advantage of this method is that the monitoring of links is already implemented in existing network hardware; no change or update of hardware is required. Another privilege is that the overhead caused by this type of measurement is foreseeable: As each link needs to be monitored, the overhead is a function of the number of links, which typically remain very constant.

But there are also drawbacks to be aware of. For link-based monitoring, monitoring parameters are chosen per link. This prevents an application- or user-specific monitoring, where monitoring parameters are chosen to fit the characteristics of an application or the needs of a user.

In order to calculate end-to-end performance parameters, it is necessary to know which path the packets took through the network. This is a non-trivial task. Link failure, dynamic

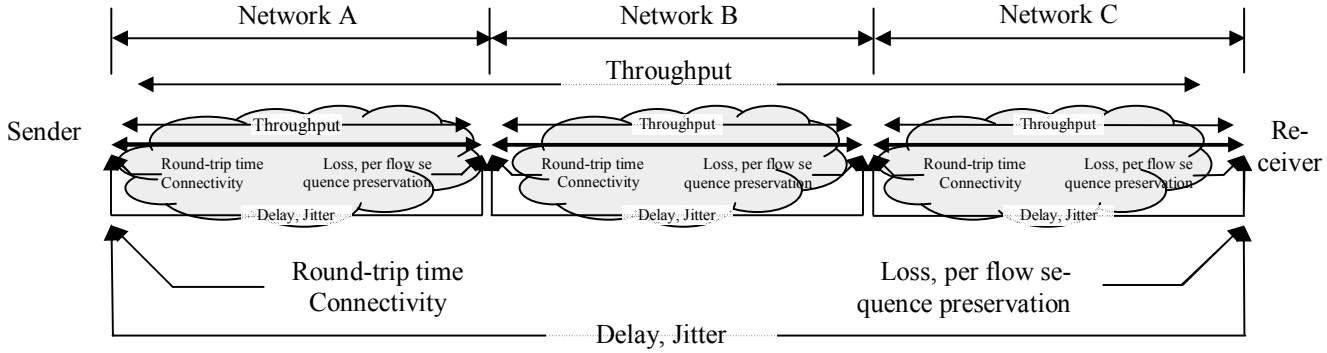


Figure 2. Measurement for multi-domain auditing

routing, or load-balancers might lead to non-deterministic routing decisions requiring to asset the concrete path on a regular basis. The Trial-Balloon Procedure [5] defines one mechanism, which is able to determine the path of a flow.

Another possible approach is a flow-based measurement. For the notion of a "flow" several definitions exists. A very broad definition by the IP Flow Information Exchange (IPFIX) working group can be found in [22], which groups packets according to a set of common header fields. While this definition is valid from the perspective of IPFIX, for auditing it is too coarse-grained. Here a flow is defined as an application-level end-to-end stream, where all packets exhibit a common traffic profile. The significant difference of both definitions is that the latter does not allow for aggregation and it is independent of any fields of the IP header.

With a flow-based measurement, probes are included in a user's flow with the same address information. They, therefore, take the same route through the network as the user data, independently which one it is. The problem finding the correct path is solved by the network itself.

As another advantage user and application requirements can be taken into account when generating probes, *e.g.,* during a Voice-over-IP session with silence suppression, no or less probes are generated during periods of silence.

A disadvantage of this approach is that the auditing overhead is not foreseeable for a provider. For flow-based measurements the number of probes and, therefore, the overhead is a function of the number of flows under audit. This number is highly dynamic and difficult to predict. Furthermore, especially in backbone networks, there will be more than one flow with a turned on auditing function on one link. This doubles the work as the information about link conditions is already determined by one set of probes.

Weighting drawbacks and advantages of both approaches presented, shows that the management of a flow-based measurement is simpler and application- and user-specific requirements can be taken into account. Discovering the path of a flow is almost as costly as the measurement itself, taking into account that the path discovery has to happen on a regular basis in order to spot path changes quickly.

But the most significant advantage of a flow-based measurement approach is the possibility to shift load from the network to end-hosts, helping to improve scalability [23]. This topic will be discussed in more detail in the following section.

B. Management of Measurement Data

Within a measurement scenario, two general types of data can be observed, which are caused by measurement: signaling and probing traffic. Signaling traffic contains information, which controls the measurement process, i.e. starting and stopping of sessions, or negotiations about measurement parameters. The control part of the *One-Way Active Measurement Protocol* (OWAMP, [24]) termed OWAMP-Control is an example for such signaling traffic.

The second type is probing traffic. As the name already suggests, it consists of probes used when probing techniques are applied. In general this is a series of packets, where the type of packets, packet size, packet frequency, and duration of the series depends on the measurement algorithm and most probably on parameters negotiated. The payload of these packets is mostly without any meaning. Some algorithms include a sequence number, a timestamp, or addressing information in the payload, but these occupy only a fraction of the space available. Any remaining space is filled with a random sequence of bytes, in order to impede potential compression.

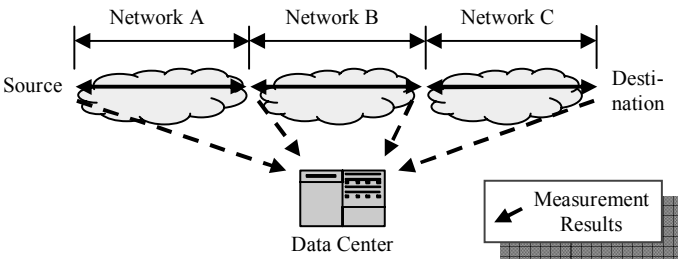


Figure 3. Measurement data transfer

If those results measured need to be processed further or archived, they must be transferred as signaling packets to data centers. This may produce a high overhead in the network and a bottleneck on the data center, because a distribution of this load on different data centers is often not possible (*e.g.*, one center may need to hold all information on one flow in order to calculate all delays required).

In this work, the separation of probing and signaling traffic is called *out-of-band measurement*[2]. The name "out-of-band" is chosen, because the stream of data contained in these probes can be seen as one "band". The stream of signaling data is not contained in this stream, therefore, it is "out-of-band".

In order to avoid the disadvantages mentioned above *in-band measurement* is proposed. Following the definition from the last section, in-band measurement would mean, carrying all information in one single band, i.e. in the payload of probes.

This avoids additional signaling traffic, thus, it reduces the overhead significantly. This is even more important for auditing. As illustrated in Fig. 3 and in order to locate violations, intermediate results from borders of each administrative domain need to be transferred to a data center for further evaluation. "Data center" here stands for the role of a system, which evaluates measurement results. This role may be taken up also be several systems, which share the evaluation process.

On the first glance, sending signaling information within the data stream seems to be unreasonable. As it can be seen in Fig. 3, the paths of signaling traffic and probing traffic are different; measurement results are reported to the data center and probing traffic is transferred to the destination host. Sending measurement information in the payload of data packets would mean that all measurement results will arrive at the destination only.

The remainder of this paper develops the protocol which makes use of in-band measurement to support auditing. The protocol is designed to work in multi-domain environments, and supports passive, but especially probing measurements.

[2] The term "out-of-band" and the its opposite "in-band" can be found in the measurement context with different meaning, too (see, *e.g.*, [1]). If Quality-of-Service (QoS) mechanisms are in place, out-of-band measurement can mean sending probes within a different traffic class; in-band means sending probes within the same traffic class valid for the data themselves.

C. Principle Operation and Characteristics

MeSA uses probing to measure delay and jitter. A sender periodically intersperses MeSA probes in the application packet stream, i.e. a flow as defined in Subsection A. Each *MeSA-enabled router* (or simply *MeSA router*) appends a time-stamp to these probes, whenever the packet is received, before queueing the packet for sending. The receiver can calculate the delay between each two MeSA routers from these timestamps. Not all routers need to be MeSA-enabled. In case of auditing, only per-domain results are of interest. Therefore, only at borders of these domains MeSA routers need to be established. Thus, the ISP autonomy of a domain to be traversed has been achieved.

Fig. 4 shows a sample scenario, where two end-systems are connected via two providers (ISP A and ISP B). These ISPs run operational MeSA routers at their borders. Besides "normal" application data and their respective packets, the sender periodically generates MeSA probes. MeSA routers append timestamps to these probes and, finally, the receiver evaluates these probes and calculates delays. In the lower part of the graphic it is shown, how one probe evolves over time. The probe grows with each MeSA router it traverses by one entry[3]. The time the probe spent in each domain is calculated by subtracting the timestamps.

This approach is tailor for auditing, finding a tradeoff between measurement error and effort. Errors could occur, because processing delays in routers are not considered and the synchronization of the clocks of MeSA routers might be imprecise, especially, when synchronization is performed over the network. The impact of the first issue is tolerable, since the processing delay is several magnitudes smaller than typical SLO thresholds, some 10 µs versus some 10 ms.

Concerning the second issue, the Network Time Protocol (NTP) is designed to synchronize system clocks to the accurate time as close as possible. The NTP software is available for almost every workstation and server operating system, and it is implemented in nearly every router and switch. The Network Time Protocol Version 3 [15] has provides nominal accuracy in the lower range of milliseconds, The new NTP Version 4 [16] is expected to improve accuracy by a factor of ten [17], thus, the synchronization error is also tolerable.

D. Phase 1 "Negotiation"

An auditing session starts with the negotiation phase. Its task is, firstly, to authorize the user for requesting and running auditing services, and secondly, to supply all parties with necessary information. The overhead introduced by this phase has to be minimal; especially a long latency time before the data transfer can start has to be avoided.

Authorization is done by edge routers, to which users are connected to. On requesting a new auditing session, routers forward the request to an authorization server, which looks up the user's profile and determines the user's authorization to use auditing. In case of a failure the server returns an error

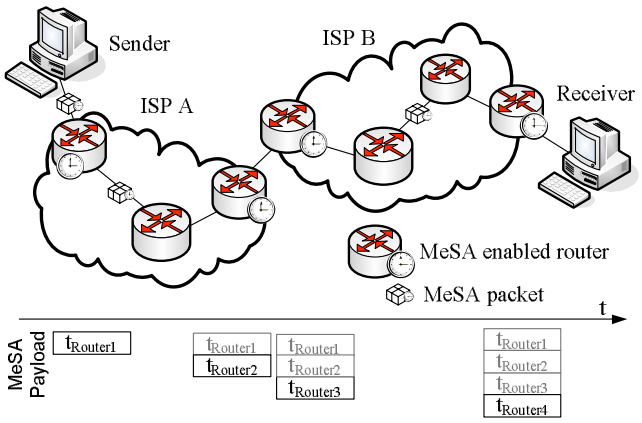


Figure 4. MeSA principle operation

message, in case of success a session identifier (SID). The SID is unique for the forwarding edge router and for the lifetime of a session. The edge router stores the SID in an authorization table and sends it to the user. The user is supposed to include the SID in every MeSA probe, and the edge router will check, if MeSA probes include a valid SID.

The MeSA protocol stores state information only in the sender, the receiver, and, if required, in the edge router of an ISP, the sender is connected to. No state is stored in intermediate systems in order to avoid scalability problems.

E. Phase 2 "Data Transfer"

During the lifetime of a flow, which shall be audited, the sender periodically intersperses MeSA probes in the data stream. MeSA probes are IP packets, which share the same IP header as the data packets- with two exceptions:

1) MeSA probes set the Router Alert Option. The Router Alert Option is defined in RFC 2113 [11] and RFC 2711 [20] for IPv4 and IPv6, respectively, and enable the notification of intermediate routers that a packet needs to be analyzed and processed before being forwarded. This option permits a MeSA router to quickly identify MeSA probes in the packet stream by only processing the IP header fields.

2) For IPv6 the probes must have a different flow label, as required by RFC 2460 [3].

A MeSA probe consists of

- an IP-Header with Router Alert Option (see above)
- a *MeSA Header*
- *MeSA Entries* from each MeSA router traversed.

The MeSA header is depicted in Fig. 5. The first four bits show the MeSA protocol version (Ver), followed by eight bits which are reserved for future use (Res). The SID (Session Identifier) field holds the SID in terms of 16 bits, assigned by the ISP during the negotiation phase. The SeqNo (Sequence Number) field with 16 bits is incremented by one for each MeSA probe sent and allows for the detection of the loss of probes. In order to link MeSA probes to the application data, each MeSA probe holds the flow label and the sequence number of the last data packet sent in the Flow Label field and the Appl SeqNo (Application Sequence Number) field, respectively. This sequence number, in terms of 32 bits, has to

[3] An IP packet with 1500 bytes can hold up to 48 entries with 30 bytes from MeSA routers. Given that an Internet packet on average only traverses approx. 16 routers at all [19], this should be sufficient for any healthy Internet path.

be provided by the transport-layer protocol the application data is transferred with (*e.g.*, TCP or RTP).

Figure 5. MeSA protocol header

The format of a MeSA Entry appended by each MeSA router traversed is shown in Fig. 6. In the current stage Bit 0 of the Flag field F shows if a signature is present. The rest of seven flag bits are unused. Furthermore, 8 bits of the reserved field (Res) are reserved for future use.

The ASN is the Autonomous System Number of the operator, which is assigned by the Internet Assigned Numbers Authority (IANA) [8]. The RID (Router Identifier) identifies one router within one Autonomous System. The assignment of the RID number is performed by the operator and does not need to be public. Again, the ISP autonomy is achieved. In the timestamp field (8 bytes) the router stores its current time as milliseconds past midnight 1970/1/1, a format which is usual on many Unix systems. Each entry can be signed optionally, in order to detect malicious manipulations. Thus, a 16 byte Signature field has been integrated.

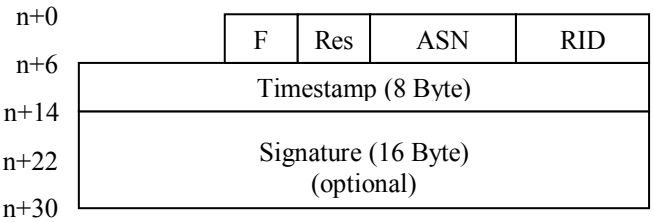


Figure 6. MeSA entry

The probes included by the sender during the data transfer phase are evaluated at the receiver. The timestamps in the probes, and the order and time of arrival of the probes enable the receiver to calculate one-way delay and jitter. Probes which show SLA violations are stored for later complaint at the operator. Throughput, loss and per-flow sequence preservation are measured in a passive manner by the receiver.

F. Phase 3 "Tear down"

After the data transfer is completed, the session needs to be torn down. Besides the release of all state information, the auditing results are transferred to the initiator of the session. Only the initiator of an auditing session, as the one who was authenticated and authorized, can complain SLA violations at his operator. Whether the sender or the receiver is in the role of the initiator depends on the traffic profile of the used application. In case of the receiver being the initiator, no data needs to transferred, as the data is already with the initiator. In the other case the auditing results are sent back to the sender.

Each ISP operates a complaint desk, where SLA violations can be reported to. The complaint is sent automatically, including the SID and the probes, proofing the violation.

V. EVALUATION

The MeSA protocol and router extensions have been evaluated by simulation and a first prototypical implementation.

A. Simulation Results

First simulations have been performed using ns-2 [18]. A scenario has been setup with 18 MeSA enabled border routers and 10 domain-internal routers in 6 domains. Fig. 7 shows the scenario without detailing all those routers. Twelve systems are placed in the scenario generating background traffic, which is not measured. The end-system labeled "Start" is used as a sender and the end-systems a, b, and c are used as receivers for MeSA flows.

The MeSA protocol was not implemented as an agent for ns-2, instead the protocol's algorithms were applied to trace files generated by the simulator. This has the advantage, that runs with different parameters can be performed on the same data easily.

Simulations have been performed to determine the impact of different sampling periods on measurement accuracy. Further simulations are used to evaluate algorithms to eliminate clock synchronization errors ex post. With the low fraction of non MeSA-routers in our scenario results have been promising.

B. Prototypical Implementation

An implementation architecture for the MeSA protocol is shown in Fig. 8. It consists of several modules, deployed on respective systems: sender, edge router, MeSA router and receiver. In this figure the sender is the initiator. For the receiver being the initiator the edge router has to be deployed in front of the receiver.

The *Initiator* modules on the sender and receiver establish new auditing sessions. The *Access Controller* on the edge router authenticates and authorizes the request. MeSA probes are generated by the *Probe Generator*. This can happen periodically or it may be triggered by the application, if the application features auditing support. The edge router checks each probe for is validity and the *Probe Stamper* appends a timestamp to each probe. On the receiver the *Auditor* evaluates probes as described in Section IV.E. The *Complaint Handler* transfers auditing results to the initiator and complains SLA violations at the ISP's complaint desk (not shown).

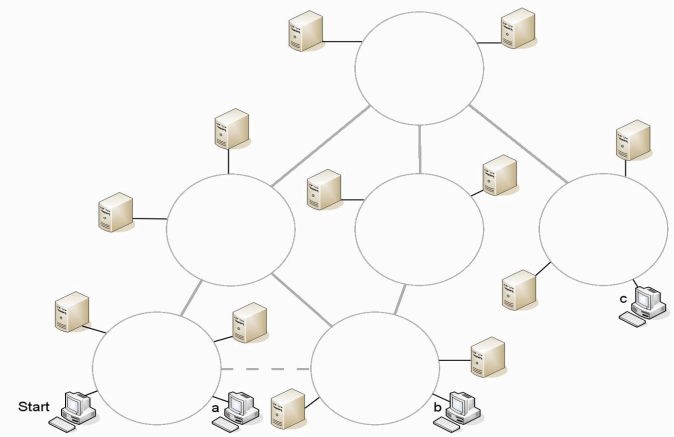


Figure 7. Simulation scenario

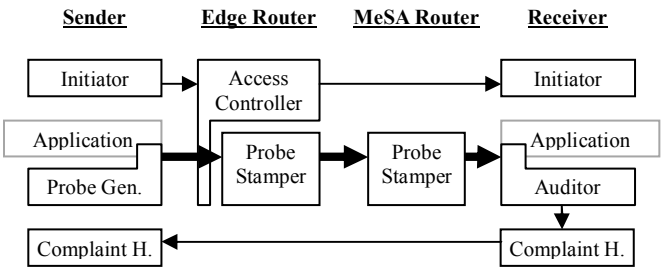


Figure 8. Auditing System Software Architecture

The prototype already features the Probe Generator module on the sender and the Probe Stamper module on the routers. A basic Auditor module on the receiver filters the probes out of the data stream and stores them for offline processing.

For the prototypical implementation, Linux systems have been used as sender, receiver, and routers. IPv6 was chosen for the application and probing traffic. We identified the implementation of the Router Alert Option as a major implementation problem. Reasons are minimal and wrong documentation respectively[4].

As another obstacle, the Linux developers allow the Router Alert option only for raw sockets. This undocumented fact requires manual processing of higher level protocol on routers.

VI. SUMMARY AND OUTLOOK

This paper presented a protocol in support of auditing, especially of IP-carrying SLAs. As a first step different measure methods have been analyzed and evaluated for their appropriateness to measure performance parameters of SLAs. A concise matrix has been sketched, showing the relationship of measurement method, performance parameter and measurement location. Hereupon the new MeSA protocol (Measured Signaling for Auditing) has been designed, combining the most suitable mechanisms. Simulation results and a first prototypical implementation have been described.

Because of limited space in this paper security related aspects of the MeSA protocol are not discussed. However, measures have been defined in order to authenticate and authorize auditing requests, to ensure the authentication, integrity and confidentiality of the data transferred in the negotiation and tear-down phase, as well as to ensure the authentication and integrity of the data in the MeSA probes. Due to the distributed nature of the MeSA protocol the last issue requires that each router has to sign its MeSA entry.

In the next phase of this work more detailed simulations will be undertaken with the help of topology generators. This will raise significance of those initial results achieved.

REFERENCES

[1] Breslau, L.; Knightly, E.W.; Shenker, S.; Stoica, I. & Zhang, H., Endpoint admission control: architectural issues and performance, SIGCOMM '00: Proceedings of the conference on Applications, Technologies, Architectures, and Protocols for Computer Communication, 2000, ACM Press, New York, NY, USA, pp. 57--69.

[2] Calhoun, P., Loughney, J., Guttman, E., Zorn, G., Arkko, J.; "Diameter Base Protocol", RFC 3588, Sept 2003

[3] Deering, S., Hinden, R.; "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, Dec 1998

[4] Engel, F.; "The role of service level agreements in the internet service provider industry", Int. J. Netw. Manag., John Wiley & Sons, Inc., 1999, 9, 299-301

[5] Eyermann, F., Racz, P., Schaefer, C., Stiller, B., Walter, T.; „Distributed Accounting and Auditing for Multiple Mobile Network Operators: Architecture" DoCoMo Euro-Labs Internal Technical Report I-ST-012, April 2005.

[6] Filsfils, C., Evans, J.; Engineering a multiservice IP backbone to support tight SLAs, Computer Networks, Volume 40, Issue 1 , September 2002, p. 131-148

[7] Hasan, H., Stiller, B.; A Generic Model and Architecture for Automated Auditing, 16th IFIP/IEEE International Workshop on Distributed Systems: Operation and Management (DSOM 2005), Barcelona, Catalunya, Spain, October 24-26, 2005, pp 121-129

[8] Hawkinson, J., Bates, T.; "Guidelines for creation, selection, and registration of an Autonomous System (AS)", RFC 1930, March 1996

[9] Ho, K., Howarth, M., Wang, N., Pavlou, G. and Georgoulas, S.; Two approaches to Internet traffic engineering for end-to-end quality of service provisioning, Next Generation Internet Networks, April 2005, pp 135- 142

[10] IETF IP Performance Metrics working group, http://www.ietf.org/html.charters/ippm-charter.html

[11] Katz, D.; "IP Router Alert Option", RFC 2113, Feb. 1997

[12] Keller, A., Ludwig, H.; "The WSLA Framework: Specifying and Monitoring Service Level Agreements for Web Services", Technical report RC22456, IBM Research.

[13] Mahajan, R.; Practical and Efficient Internet Routing with Competing Interests, Ph.D. Dissertation, University of Washington, Dec. 2005

[14] McGregor, C., Kumaran, S.; "Business process monitoring using web services in B2B e-commerce", Parallel and Distributed Processing Symposium. (IPDPS 2002), Fort Lauderdale, Florida, USA, April 15-19, 2002, .pp 219-226

[15] Mills, D.; "Network Time Protocol (Version 3) Specification, Implementation and Analysis", RFC 1305, March 1992

[16] Mills, D., Plonka, D., Montgomery, J.; "Simple network time protocol (SNTP) version 4 for IPv4, IPv6 and OSI", RFC 4330, Dec 2005

[17] Mills, D.; "Network Time Protocol (NTP) General Overview", University of Delaware, http://www.cis.udel.edu/~mills/ntp.html, visited July 2006

[18] ns-2 project homepage, http://nsnam.isi.edu/nsnam/index.php/Main_Page, visited Feb 2007

[19] Packet wingspan distribution, http://www.nlanr.net/NA/Learn/wingspan.html, visited Feb. 2007

[20] Partridge, C., Jackson, A.; "IPv6 Router Alert Option", RFC 2711, Oct. 1999.

[21] Paxson, V., Almes, G., Mahdavi, J., Mathis, M.; "Framework for IP Performance Metrics", RFC 2330, May 1998

[22] Quittek, J., Zseby, T., Claise, B., Zander, S.; "Requirements for IP Flow Information Export (IPFIX)", RFC 3917, October 2004.

[23] Saltzer, J., Reed, D., Clark, D.; "End-To-End Arguments in System Design", ACM Transactions on Computer Systems 2(4), 277-288, 1984

[24] Shalunov, S., Teitelbaum, B., Karp, A., Boote, J., Zekauskas, M.; "A One-way Active Measurement Protocol (OWAMP)", work in progress, Feb 2006

[25] Texas State Library and Archive Commision, Glossery; http://www.tsl.state.tx.us/ld/pubs/compsecurity/glossary.html, visited Feb 2007

[26] Tychon, Emmanuel; Advanced Performance Measurement with Cisco IOS IP SLA, Cisco Systems, http://www.etychon.com/presentations/Advanced_IPSLA.pdf, visited Feb 2007

[27] Zseby, T., Zander, S., Carle, G.; "Policy-Based Accounting", RFC 3334, October 2002

[4] The documentation of the Linux C API describes a wrong prototype for the creation of sockets which handle packets with the router alert option set on the router.

Frameworks for Business-driven Service Level Management

A Criteria-based Comparison of ITIL and NGOSS

Thomas Schaaf
Munich Network Management Team, Ludwig-Maximilians University (LMU)
Oettingenstr. 67, 80538 Munich, Germany
E-mail: schaaf@mnm-team.org

***Abstract*—In the majority of today's IT organizations, Service Level Agreements (SLAs) are an important means for underpinning IT service provisioning by clearly defined Quality of Service (QoS) parameters as well as service costs and violation penalties. The management of those SLAs is the main subject addressed by the discipline of Service Level Management (SLM) which covers several activities vital for the deployment of customer-oriented, high-quality and well-performing IT services.**

This paper analyzes and compares two of the most important SLM frameworks available in business-driven IT Management: the IT Infrastructure Library (ITIL) with its SLM reference process and the NGOSS SLA Management Handbook. In order to deliver significant and helpful results, we derive a set of evaluation criteria from a realistic IT scenario. These criteria are applied to ITIL and NGOSS in order to elaborate possible areas of conflict as well as complementary fields and unaddressed issues. The results are visualized in an analysis matrix which shows whether and how ITIL and NGOSS may co-exist as SLM frameworks in one operative and management environment.

I. Introduction

Service Level Management (SLM) is often regarded as one of the most important management disciplines in IT Service Management (ITSM), vital for customer-orientation and provision of high-quality IT services. SLM is responsible for determining, monitoring and reporting IT service quality metrics (QoS parameters) in line with the commercial business goals of the entire organization. It is important for an improved relationship between a service provider and its customers, because – at the best – a common understanding of expectations and possible achievements to the agreed costs is established between provider and customer. Thus, SLM builds the interface between an IT organization and its customers and therefore plays a quite decisive role in the context of business-driven IT management.

Having gained this status and attention, various concepts for supporting effective Service Level Management have evolved from research and practice throughout the last couple of years. The most important operative instrument which all of these approaches have in common, are *Service Level Agreements* (SLAs). However, the approaches differ strongly in their scope, their level of detail, their feasibility for technical and tool support and their target audience. This heterogeneity may turn out as problematic when IT managers try to implement SLM: On the one hand, a holistic approach does not exist, and on the other hand the existing concepts, frameworks and technologies do not fit together like pieces of a puzzle. Guidance in integrating multiple efforts into one consistent "SLM solution suite" is not available today.

These considerations build the starting point for the analysis presented in this paper in which we compare two of the probably most popular existing SLM frameworks by using a set of significant evaluation criteria. These two frameworks are the *IT Infrastructure Library* (ITIL) [1] and the *NGOSS SLA Management Handbook* [2] (for the remainder of this paper referred to as the "NGOSS Handbook"). Both frameworks claim for themselves to be business-aligned.

While the NGOSS Handbook is clearly focused on SLM issues only, ITIL is not. In fact, ITIL provides guidelines ("best practices") for the entire field of ITSM. Service Level Management is one of the five reference processes described in the Service Delivery book [3]. The goal of the comparison presented in this paper is on the one hand to elaborate those areas where ITIL and the NGOSS Handbook can be regarded as complementary and on the other hand to find potential fields of conflict when trying to co-implement ITIL and NGOSS SLM. Finally, the comparison shall uncover those fields in SLM for which none of the compared approaches provides a solution. For that stake, the remainder of this paper is structured as follows: Section II introduces the analysis by defining a set of fundamental terms and concepts in the area of SLM. The rest of the paper will base on these definitions and, where necessary, state differences and extensions which are made by the respective approach. In Section III, we derive evaluation criteria for SLM frameworks from a realistic IT scenario. Section IV gives a survey of ITIL SLM and the NGOSS Handbook, followed by the actual comparison whose results are visualized and explained in Section V. Further related work is presented in Section VI. The paper concludes with a short summary and outlook on future efforts.

II. Terminology & Common Concepts

In the area of SLM, various terms and concepts have been established over years and are today shared between different approaches. Although – as almost everywhere – a uniform terminology for SLM does not exist, the following set of terms is used in the majority of the presented approaches

2nd IEEE/IFIP Int'l WS on Business-driven IT Management (BDIM 2007), Munich, Germany, May 21, 2007

in basically comparable meanings. In this section, we give definitions generic enough to build the foundation for both ITIL and NGOSS:

1) Service Level Agreement (SLA): A Service Level Agreement is a written contract between a service provider and a service customer/subscriber. It must contain a description of the service functionality, definitions of related QoS parameters (service levels) and declarations of responsibilities of both parties. It may additionally contain prices for service usage to pay by the customer/subscriber and penalties for service level violations to pay by the service provider.

2) Service Catalogue: A Service Catalogue contains definitions of standard services as well as documentations of customer-specific services. It can be used as a foundation for automated service subscription or for the negotiation of SLAs.

3) Service Model, Life Cycle and Domains: When talking about SLM and SLAs, there should exist a common understanding of what the term service means. Accordant to [4], a view on a service consists of two components: the service life cycle which displays the *dynamic behavior* of a service, and the *static service model* which describes the composition of basically entities and interactions inside a service and shows the service in a role-based context.

Starting with the service life cycle model, a division of the life cycle into seven phases has proved as a reasonable scheme. These phases are: offer, negotiate, implement and test, accept, operate, change and decompose.

An extract of the static service model which has been developed in [5] is depicted in Figure 1. The relevant domains for the service context are the *provider domain* and the *customer domain*. The provider domain comprises all of the entities vital for providing the specified service functionality. The service provider is responsible for the task of service provisioning and therefore operates a service implementation and a service management.

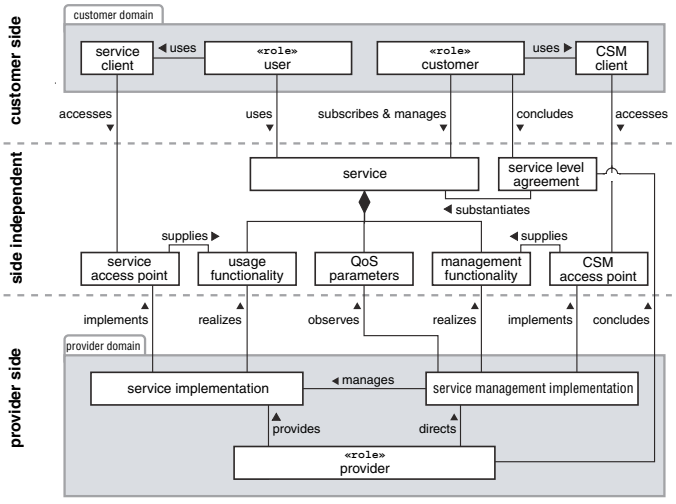


Fig. 1. Static Service Model

The customer domain contains the customer and the user role. The user can deploy the usage functionality of the provided service via a *Service Client* (SC) which is connected to a *Service Access Point* (SAP). The customer subscribes the service, concludes an SLA with the service provider and monitors service provisioning via the Customer Service Management (CSM) access point. Further on, the model defines functionality classes and several interfaces for management and usage which we do not introduce in more detail at this point.

4) Service Availability and Reliability: Having defined the SLA and Service Catalog concepts as well as a Service Model covering dynamic and static characteristics of IT services, we finish this section by introducing some of the probably most important IT service performance indicators: availability, mean time to repair (MTTR), mean time between failures (MTBF) and reliability.

Provided that T_{op} is the agreed service operation time and T_{down} is the (cumulative) service downtime, the (predicted, agreed or actually measured) availability of an IT service is defined as:

$$Availability = \frac{T_{op} - T_{down}}{T_{op}} \tag{1}$$

In practice, determining service availability may be much more complicated than this simple formula suggests, due to the complexity of most IT services and the complexity of the measurement process. The NGOSS Handbook addresses this issue by introducing Service Degradation Factors and Service Access Point Weighting which we shortly explain later on in Section IV. A doctoral thesis presenting a methodology for the determination of service availability can be found at [6].

Two common metrics in SLAs are the MTTR which is the average duration of a service incident, and the MTBF, defined as the average service uptime without interruption. Provided that n is the number of incidents within the considered time period, t_0 is the start time of this period, $t_{i,down} (1 \leq i \leq n)$ is the occurrence time of the i-th service outage, $t_{i,up}$ its clearing time, and t_{n+1} is the end time of the period, the MTTR is defined as:

$$MTTR = \frac{\sum_{i=1}^{n} |t_{i,down} - t_{i,up}|}{n} \tag{2}$$

Analogous, the MTBF is defined as:

$$MTBF = \frac{(t_{n+1} - t_0) - \sum_{i=1}^{n} |t_{i,down} - t_{i,up}|}{n+1} \tag{3}$$

Knowing MTTR and MTBF, availability can also be calculated using the following formula

$$Availability = \frac{MTBF}{MTBF + MTTR} \tag{4}$$

which will deliver the same values for availability as the first definition. Finally, the reliability of an IT service is an indicator for the frequency of service incidents/outages. A high MTBF results in a high reliability. Compared to availability, MTTR and MTBF, reliability is a weaker performance indicator, hard to express by an intuitive formula. Nevertheless,

it is often not the service availability, but its reliability that is responsible for a customer's subjective satisfaction and the resulting provider's reputation. One goal of SLM is to maximize service availability and service reliability.

III. Analysis Foundations

A thorough and structured analysis and comparison of the given frameworks requires meaningful evaluation criteria which we derive from a typical SLM scenario presented below. Before this, we give a short motivation for our choice of frameworks to compare.

A. Why compare ITIL and NGOSS?

ITIL is a today widely-used collection of best practices in IT Service Management that has, of all standardization efforts, gained the biggest popularity. It builds the foundation for the ISO/IEC 20000 standard [7]. SLM is one of the topics addressed by ITIL and at the same time one of the ten ITIL reference processes. By contrast, the NGOSS SLA Management Handbook is by far the most comprehensive and voluminous published collection of SLM concepts and principles. In addition, both ITIL SLM and the NGOSS Handbook fulfill the following characteristics that we regard as specific and determining for any approach that can be called a "framework": They can be regarded as holistic (i.e. not restricted to specific aspects, but addressing SLM "as a whole"), they use and partially define their own terminology for SLM, and they are independent from specific tools.

B. A typical SLM scenario

As depicted in Figure 2, we consider an exemplary IT provider P that provides three different IT services (E-mail, Webhosting and Backup) for its two customers C1 and C2. While Webhosting is exclusively delivered to C1 and Backup exclusively to C2, the E-mail service is offered to both customers. Accordant to the generic Service Model presented in Section II, all services are accessed by the users of the respective customer domain via a customer-specific Service Client (SC) which is connected to the Service Access Point (SAP) of the respective service.

To make its three services available, P is dependent from two external providers, the suppliers S1 and S2. S1 may be considered as a typical Telecommunication Provider, S2 as a company specialized on the implementation and operation of individual application services. Both S1 and P require one of S2's services (the "Application X" service). Of course, P as well as S1 and S2 host their own IT infrastructure represented by the clouds. We assume that Service Clients are only needed when human users access an SAP. If services are needed as sub-services and thus form components of another service, they can be directly connected via the SAP. SLAs have been closed for all services, although we have only plotted two of them exemplary.

We selected this scenario since it offers the following characteristics: It shows a service provider in the context of multiple customers as well as multiple external suppliers (multi domain scenario), providing a set of services that are assembled of IT components and sub-services. For an SLM framework, this is both an authentic and challenging use case.

C. Assessment Criteria for SLM frameworks

The following set of evaluation criteria for SLM frameworks has been derived from the above SLM scenario. What should an SLM framework provide to an IT manager or Service Level Manager in order to facilitate effective SLM? We present each criterion in the following structure: First we describe and explain it, then we show how this criterion can be derived from or motivated by the scenario. Every criterion is assigned to one of the following three categories.

Management Aspects:

M1 **Management process** Following the principle of process-orientation, any incurring task in the area of effectively managing SLAs should be embedded within an embracing management process. A management process is characterized through a well-defined sequence of activities with clearly delineated responsibilities for every step or task. A framework for SLM should specify the SLM process, its activities, corresponding responsibilities and process in- and outputs.
In the scenario: The SLM process of the provider P is responsible for negotiating, establishing and monitoring all SLAs with C1 and C2 as well as S1 and S2.

M2 **Relationships and Dependencies to other management disciplines** SLM is not a management function which acts in isolation to any other management discipline. The opposite is the case. That is why an SLM framework should be aware of its direct management environment and – at the best – define interfaces for the communication within this environment.
In the scenario this becomes visible when effective SLM for example depends on certain outputs/data coming from Configuration Management.

M3 **Management assessment guidelines** The degree to which the process (as is the case when looking at ITIL) or the tools and recommendations (NGOSS) perform in a specific use case should be assessable and measurable. Therefore, the framework must give concrete advice on how to evaluate its own implementation. Critical Success Factors (CSF) and Key Performance Indicators (KPI) are required.
For the stake of continuous improvement, P should review its own SLM process at regular intervals.

M4 **Business alignment of SLM** Whether recommendations or decisions that an SLM framework helps to make are sufficiently in line with business needs, is hard to determine, since business needs may vary to a great extent in different scenarios. An SLM framework can be regarded as business-aligned, if significant decisions consider business impact which is basically expressed by monetary loss or return.
This is especially important when C1 and C2 are internal customers, i.e. P, C1 and C2 stand under one

administrative business domain.

SLA and QoS aspects:

S1 **Mapping support for QoS parameters** The agreed service performance has to be quantified in terms of QoS parameters. Therefore, service quality metrics (e.g. the availability of an IT service) have to be broken down on physical resource performance metrics (e.g. router availability). This is important for the service provider in order to avoid performance promises that he will not be able to fulfill – for example due to hardware restrictions. An SLM framework should give advise on how to proceed to vertically map QoS parameters between services, sub-services and resources.
In the scenario: P must know which performance metrics for the E-mail, Webhosting and Backup services he is able to promise to C1 and C2 within the SLAs.

S2 **Measuring support for QoS parameters and service performance** QoS parameters need to be measured. In most cases, the measurable units within an IT environment are not the same ones as the ones specified in an SLA. Often, the SLA-relevant metrics are aggregations of physically measurable performance metrics. However, an SLA framework should provide support in measuring and aggregating QoS parameters.
P must be able to measure and aggregate the QoS metrics made available by its own infrastructure as well as the performance metrics of the sub-services purchased from S1 and S2.

S3 **SLA templates or design rules** An essential task in SLM is the establishment of the SLA documents including negotiations with all customers. An SLM framework should support this task by providing customizable templates for SLAs or guidelines for contract design.
P must establish SLAs with C1 and C2 for all delivered services.

S4 **Performance calculation and reporting support** To calculate and report amongst others the achieved service levels, the degree of service degradation and the number of SLA violations is an essential task of SLM. A framework should provide support on this issue. Reporting and QoS determination also build the foundation for service charging which is addressed by the next criterion.
P must be able to calculate the performance of its three services from the measured QoS metrics and create reasonable reports for its customers C1 and C2.

S5 **Support of SLA-based charging and accounting** To charge a customer for service usage is not only a relevant need for companies specialized on service outsourcing. Charging becomes rather more important for all IT organizations – even the ones serving internal customers only – in order to strengthen the perception of internal customers as business partners. Since in

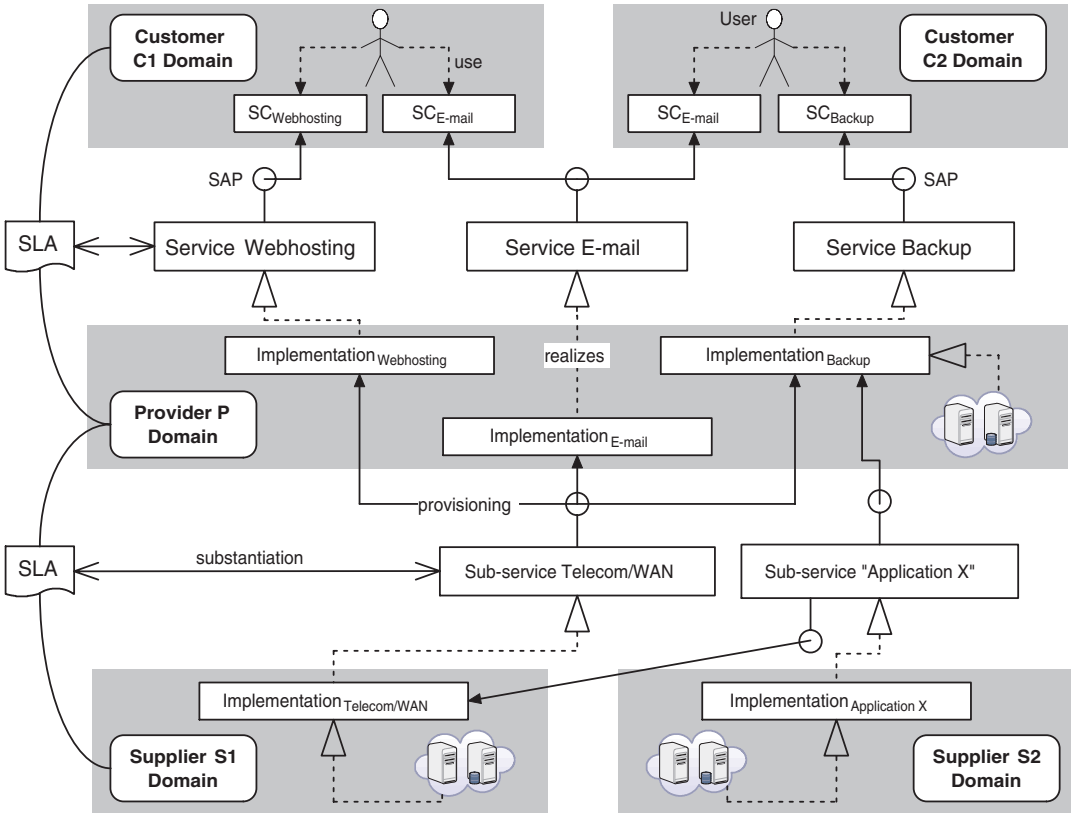


Fig. 2. Scenario

Financial Management for IT Services accountable units often correlate to QoS metrics (in many cases, they are even congruent), it stands to reason to consider charging issues already in the process of service negotiations and SLA design, and thus within the responsibility of SLM. That is why an SLM framework should give guidance on integrating charges into SLAs by e.g. considering them in SLA templates or in the SLA establishment process.
P has to charge C1 and C2 for the delivery of its three services. The SLA negotiations should cover agreements on rates and penalties which should be recorded within the SLAs.

General aspects:

G1 **Support for multi domain service provisioning** An SLM framework must provide support for those scenarios where service performance and the achievement of certain service levels highly depend on the performance of underlying services, obtained from third party providers.
P's services are not only dependent on P's own IT infrastructure, but also on the services delivered by P's suppliers. Thus, achievable service levels highly depend on the service levels achieved by S1 and S2.

G2 **Support for multiple customers service provisioning** At first sight, this criterion looks similar to the previous one, but means something different: While the previous aspect refers to linear chains of service provisioning over different providers, the multiple customers service provisioning support means that the SLM framework should give guidance on how to manage different SLAs of different customers efficiently. Modular design of SLAs and the establishment of a Service Catalogue are two possibilities for facilitating the management of multiple SLAs. One goal must be to avoid redundant pieces of work.
Both C1 and C2 subscribe P's E-mail Service. Their SLAs concerning this service may differ in some aspects – e.g. C1 and C2 may purchase different service levels. But some parts of the SLAs for the E-mail service may be exactly the same, e.g. the description of the basic service functionality.

G3 **Automation and Tool support** An SLM framework can provide automation and tool support by maximizing its degree of formalization. The adoption of modeling languages and tools like XML or UML can help to substantiate the framework contents in a formal way to facilitate development and deployment of tools.
In the scenario, tools can be a helpful addition to SLM for example at the following spots and interfaces: An automated Service Catalogue enables the customer to select between a repertoire of standard services with their default configuration. A tool for storing and administrating SLAs helps the SLM staff to manage lots of documents and avoid version conflicts. Other tools may be used to check the integrity of SLAs and find inconsistencies or to monitor process flows.

D. Weighting of Criteria

We forbear from assigning specific weights to the listed criteria. This is not necessary for the comparison, since we want to explore areas of conflicts and complementary fields in the approaches as well as issues unaddressed by both frameworks. The goal of this analysis is *not* to give a statement on whether ITIL SLM oder the NGOSS Handbook is the better (or even the best) solution.

IV. Survey: ITIL and NGOSS

In the following, we give a survey of ITIL SLM and the NGOSS SLA Management Handbook. Doing this, we keep in mind the criteria developed in the previous section to evaluate these two frameworks later on. At certain points of the descriptions, we refer to the criteria catalog in order to prepare the assessment in the next section.

A. ITIL Service Level Management

1) Overview: ITIL (IT Infrastructure Library) is a collection of books, in which best practices in IT Service Management (ITSM) are described. Today, ITIL can be seen as a de-facto standard in the discipline of ITSM, for which it provides guidelines by its current core titles Service Support [8] and Service Delivery [3]. ITIL follows the principle of process-oriented (IT Service-) Management: Every management activity taking place within an IT organization is part of one of the defined management processes. Thus, process-orientation extends the idea of functional management where IT management decisions and actions take place in different departments (e.g. network department, server department, storage department). In effect, the responsibilities for specific IT management decisions can be shared between different organizational units as the management processes span the entire IT organization independent from its organizational partition.

Service Level Management is one of the ten ITSM processes defined by ITIL and part of the Service Delivery book. Besides Service Level Management, this book describes Availability Management, Capacity Management, IT Service Continuity Management and Financial Management for IT Services. Together, these five processes are called the tactical processes and build the direct context of SLM as depicted in Figure 3 – in contrast to the operational ITSM processes described in the Service Support book (e.g. Incident Management, Change Management).

2) Roles: The relevant roles in ITIL Service Level Management are the service provider, the service customer and the service user which exactly maps the generic service model introduced in Section II. The SLM process builds the interface between the IT organization (as the service provider) and its internal and external customers. According to ITIL, any customer is characterized through the commission, payment and ownership of one or more IT services that are provided by the IT organization. Due to the roots of ITIL in the

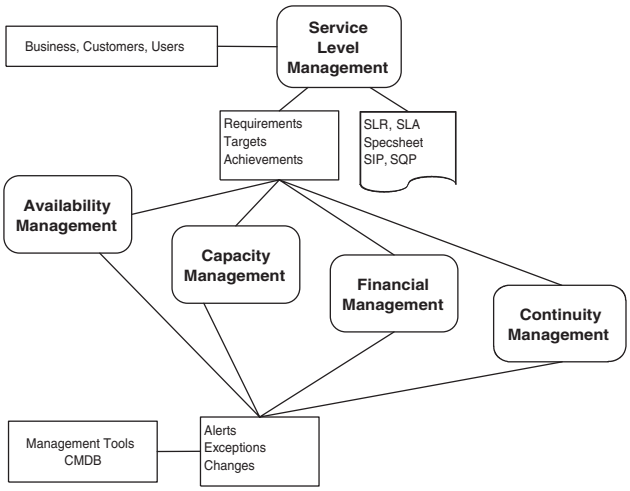


Fig. 3. Context of ITIL Service Level Management

British government, the focus is clearly set on internal – which basically means non-commercial – customers like e.g. the manufacturing or accounting department of the enterprise in which the IT organization is located. A user is defined as any person using a commissioned service (e.g. manufacturing staff or an employee in the accounting department).

SLAs are closed between the IT organization and its customers. Responsible for the contract negotiations are the process owner (Service Level Manager) who represents the IT organization, and of course a representative of the customer.

3) SLA types and structures: Before discussing the management process and its activities, ITIL enhances the SLA concept by some additional aspects and terms: Any SLA that is closed between the IT organization and at least one of its internal customers in order to provide IT services to this customer, is called an *Operational Level Agreement* (OLA). By contrast, an SLA that the IT organization contracts with an external supplier in order to obtain (sub)services from this provider, is called an *Underpinning Contract* (UC). ITIL makes this distinction, because UCs may significantly impact what can be promised within an OLA. The Service Level Manager must know about the interdependencies between the existing UCs and OLAs *(cf. criterion G1)*.

Besides these two special types of SLAs, ITIL defines three different kinds of SLA structures. An SLA is called a *Service-based SLA* when it is valid for all customers of one or more services and no individual SLAs are designed for different customers. This is often the case for services facing uniform customer/user requirements (e.g. e-mail, internet-access). By contrast, a *Customer-based SLA* is closed with one individual customer and normally for all the services subscribed by this customer. Customer-based SLAs basically find appliance when customer demands regarding service quality parameters vary to a high extent. The third SLA structure ITIL proposes is the so called *Multi-level SLA* which can be regarded as the composition of Service- and Customer-based SLAs. ITIL describes Multi-level SLAs to follow a three-layer structure:

- The *Corporate Level* covers generic issues valid for all customers (e.g. opening hours of the Service Desk).
- The *Customer Level* contains customer-specific issues as extensions to the Corporate Level, regardless of the services ordered by this customer (e.g. expanded Service Desk opening hours for some customers).
- The *Service Level* covers service-specific issues relevant for one customer or group of customers (e.g. agreed availability for an accounting service).

The idea behind this three-layer structure is to substitute only the service and customer layers in order to avoid redundancy and reduce maintenance and administration expense when designing new SLAs *(cf. G2)*.

4) The management process: There are six activities that build the ITIL Service Level Management process: Identify, Define, Contract, Monitor, Report and Review. Each of these activities needs certain inputs and creates certain outputs both of which are described by ITIL *(cf. M1)*.

1) **Identify:** First of all, the customer demands have to be identified and described as *Service Level Requirements* (SLRs). This document will provide a basis for the future SLA.
2) **Define:** In the second step, the concrete service which is to deliver to the customer, has to be defined in case of a new/individual service. Therefore, a *Service Specification Sheet* (Specsheet) should be created, containing information on the technical implementation and realization of the service. The Specsheet can be seen as a translation of the SLRs into technical specifications. Additionally, ITIL proposes to develop a *Service Quality Plan* (SQP) as an internal document containing i.a. key performance indicators and a plan for achieving the agreed service quality. In case of existing services, the respective documents can be adopted.
3) **Contract:** In this activity, a written contract between the IT organization and its customer is closed, based on SLR, Specsheet and SQP. In the case of an internal customer, this contract (SLA) is an OLA. Another output of this step is the updated *Service Catalogue*. In case of a new or modified service, the changes should be added to the Service Catalogue in order to potentially provide this service to other customers, too. The Contract activity also implies to close or update UCs with external suppliers, if the agreed service requires sub-services, technology or infrastructure which the IT organization is unable or unwilling to provide by itself.
4) **Monitor:** Of course, the actually achieved service levels (QoS parameters) have to be monitored. It is important that the SLA contains information on *how* and *how often* the measurements have to take place. Measurement outputs are called *Service Level Achievements* and serve as inputs for the next activity.
5) **Report:** In this activity, the achieved service levels are compared with the agreed service levels in order to detect violations. *Service Level Reports* are created and

handed to the Service Level Manager.

6) **Review:** As the last activity in the ITIL SLM process, the Service Level Reports are evaluated with respect to the contracted OLAs and UCs and of course under consideration of the SQP. In order to continuously improve the service quality, a *Service Improvement Program* (SIP) should be developed and launched within the next time period.

As critical factors for the success of the SLM process, ITIL names aspects such as the expertise and customer-orientation of the process manager and clearly delineated responsibilities within process execution. Key Performance Indicators (KPIs) help assessing and measuring the success and performance of the process and its outputs *(cf. M3)*. There are of course many interdependencies between the SLM process and all the other nine ITIL processes. For example, Service Level Management and Financial Management for IT Services should cooperate closely in order to integrate service charges into SLAs. At the same time, Incident Management needs to be aware of the impending violation fees in order to assign the right priorities to the incident tickets.

5) Conclusion: ITIL gives guidance for the organizational setup of an SLM process. It is focused on a clear role model, the definition of responsibilities, activities and a common terminology as well as business-awareness and -alignment of ITSM. Covering this radius, it remains superficial in many areas. It is tool-independent, only little technology-aware and does not provide concrete templates for the various artifacts (even not for an SLA). The NGOSS SLA Management Handbook, presented in the following, gives more detailed guidance in some fields.

B. NGOSS SLA Management Handbook

1) Overview: The SLA Management Handbook is a publication of the Tele Management Forum (TMF) and part of the New Generation Operation Support Software (NGOSS) project [9]. It consists of four volumes named Executive Overview [2], Concepts and Principles [10], Service and Technology Examples [11] and Enterprise Perspective. The fourth volume (Enterprise Perspective) has not been released to public yet[1].

These titles address different parties of interest: Volume 1 (Executive Overview) has been written for Chief Executive Officers, Volume 2 (Concepts and Principles) for telecommunication managers, Volume 3 (Service and Technology Examples) for telecommunication implementers, and Volume 4 (Enterprise Perspective) will primary address enterprise managers.

Although the contents of the NGOSS Handbook are basically aligned to businesses in the telecommunication industry, they are generally also applicable to a broader scope of IT-dependent organizations, because most of the presented concepts are not restricted to telecoms industries specifics.

The probably most significant title of the SLA Management Handbook suite is the second one (Concepts and Principles). It starts with some *Business Considerations* including a business model, followed by a chapter on *Telecommunications Services*. The next three chapters deal with several subjects directly concerning SLAs, starting with *SLA Content and Management*, followed by *SLA Management Tools* and concluding with a chapter on *SLA Performance Reporting*. In the following, we give a survey on the contents of these five core chapters.

2) Business Considerations: In contrast to ITIL, the role model applied in the NGOSS Handbook is more diversified, particularly with respect to the existence of external suppliers in the end-to-end service delivery chain *(cf. G1)*. But the basic business model is the same as we have already seen in ITIL: Basically, an SLA is regarded as a contract between one customer and one service provider. This contract is located at the customer-provider interface. An extension of this simple model is the provider centric Business Relationship Model. It adds the roles of complementary providers, third party service providers, function and process suppliers, intermediaries and hardware/software/solution vendors.

As the biggest stumbling block in SLM, the NGOSS handbook describes the end-to-end service challenge that consists in the delivery of a seamless service through a number of trading partners which is indistinguishable from the same service provided by a single supplier. In the remainder of the handbook, this e2e service challenge is always kept in mind. The handbook even claims to be the first open document addressing e2e service performance issues in SLM.

The Business Considerations chapter finishes with a section on service measurement and performance metrics *(cf. S2)*. It first defines four *basic prerequisites* for an effective service measurement:

1) Parameters must be measurable.
2) The quantification method must be described.
3) A review of the delivered performance must take place.
4) Penalties or incentives must be specified.

Furthermore, a *metric* is defined as a measurable parameter. Ten requirements are defined that should be fulfilled by any specified performance metric. To give an impression, we exemplary name three of them:

- The metric should provide concrete repeatable measurements in well-defined quantities and without subjective interpretation.
- The metric should be useful as a specification in a contract in order to enable the customer purchasing the service level he needs.
- The measuring process should be acceptable to service providers and customers, and artificial performance goals should be avoided.

Implicitly, all of these business considerations build the requirements for the developed solutions and draw the context for them.

3) Telecommunications Services: A telecommunications service is characterized by an object-oriented model with the two central entities Service Function and Service Resource. The latter is a superclass of hard- and software, staff

[1]March 2007

and intellectual property and licenses, demanding only hard- and software as mandatory components. Service Resources however enable Service Functions which are divided into the service's Primary Function, Enabling Function and OAM Function (Operation, Administration, Maintenance).

This model is much simpler than the generic service model introduced in Section II, but not contradictory to it. It uses the same domains (customer and provider domain) and shares the concept of an SAP. New aspects are the possibility of aggregating SAPs to SAP groups and regarding different layered provider domains. An additional feature of the model is given by its *Service Elements* (SEs) that are abstract entities out of which a service is composed. In this capability, SEs can be shared between different provider domains – e.g. a service provided by A may base upon an SE out of the provider domain of B although not being completely dependent of one of B's full services. Thus, an SE can be a sub-service, a physical resource, a human resource or what ever may be used to build a service.

Furthermore, the NGOSS Handbook proposes a *Customer Contact Point Reference* (CCPR) model which serves as a model for exchanging service performance and management information between a provider and its customer. The *Customer Contact Point* (CCP) is the logical point at which a customer may manage the services he has subscribed for. In the generic service model of Section II, the CCP is given by the CSM access point which can be accessed by the customer through a CSM client. Again, the two models are very similar.

With respect to service performance, the NGOSS handbook introduces additional concepts two of which are the *Service Degradation Factor* (SDF) and *SAP Weighting*. An SDF can be used when calculating service (un)availability. It is based on the idea that besides performing well and failing completely, a service may also be partially degraded, but still usable. In order to consider this fact in availability calculations, an SDF – which can take any value between 0 (service fully available) and 1 (service fully unavailable) – is assigned to each outage event type. Provided that T_{op} is the agreed service operation time, T_{down} is the service downtime and SDF is the Service Degradation Factor for the regarded outage event type, the formula for service availability is now:

$$Availability = \frac{T_{op} - (SDF \cdot T_{down})}{T_{op}} \tag{5}$$

Weighted SAPs can additionally be useful in order to take into account different impacts of outages related to different SAPs when calculating service availability *(cf. S4)*.

4) SLA Content and Management: Based on the business considerations and the service model, the handbook gives recommendations for the concrete SLA content and design *(cf. S3)*. These recommendations are arranged in four categories which are:

1) **Fulfillment Process** (Recommendations 1 to 5): This category contains recommendations concerned with the negotiation and engineering of SLAs.
   - *Example (Recommendation 2): For any service the customer should be able to select a) parameters to guarantee and b) value ranges for the parameters.*
2) **Assurance Process** (Recommendations 6 to 13): After concluding the SLA, the recommendations in this category should be followed by the provider when delivering the service to its customer.
   - *Example (Recommendation 8): Strong access control and authentication must be provided so that customers are able to access their own data to the extent agreed in the SLA.*
3) **Customer Interface Management** (Recommendations 14 to 16): This category contains recommendations for the communication between provider and customer concerning SLAs and services.
   - *Example (Recommendation 16): The provider's CCPs should have information available on the status of any service about which the customer could inquire.*
4) **General Recommendations** (Recommendations 17 to 23): The last category contains general recommendations viable for SLM like e.g. modular assembly of the SLAs or the definition of provider and customer responsibilities.

5) SLA Management Tools and SLA Performance Reporting: Due to space restrictions, we only give a short overview of these two remaining chapters: The title "SLA Management Tools" might mislead to the assumption that this chapter deals with software tools for SLM – which is not the case. It rather presents a Service Life Cycle model which is almost identical to the one we outlined in Section II, followed by a *KQI (Key Quality Indicator) Development Methodology* that shall help to identify metrics that capture the customer's QoS perception. The third tool is the *SLA Parameter Framework* that organizes performance parameters into specific categories.

In the SLA Performance Reporting chapter, a *Performance Reporting Interface*, several sequence diagrams for different reporting scenarios as well as a *Performance Reporting Process State Model* are presented.

6) Conclusion: The NGOSS SLA Management Handbook covers much more aspects and detailed proposals vital for SLM than ITIL does. This is not surprising, since SLM is only one of a total of ten ITIL processes. While the second Volume of the NGOSS Handbook already consists of 204 pages of text, ITIL SLM is described within 33 pages in a comparable style. The descriptions in the NGOSS Handbook are less superficial and much more aimed at straight deployment.

V. Comparison

We now apply the assessment criteria from Section III to the frameworks presented in the previous section. The results are made visible in table form and summarized below.

A. Assessment matrix

The assessment matrix shown in Table I lists the three groups of evaluation criteria, and for each criterion its degree

Group	*Assessment criteria*	*ITIL SLM*	*NGOSS Handbook*
Management aspects	M1: Management process	√	×
	M2: Relationships and Dependencies to other management disciplines	√	×
	M3: Management assessment guidelines	√	×
	M4: Business alignment of SLM	(√)	(√)
SLA and QoS aspects	S1: Mapping support for QoS parameters	×	×
	S2: Measuring support for QoS parameters and service performance	×	√
	S3: SLA templates or design rules	(√)	√
	S4: Performance calculation and reporting support	(√)	√
	S5: Support of SLA-based charging and accounting	×	×
General aspects	G1: Support for multi domain service provisioning	×	√
	G2: Support for multiple customers service provisioning	√	(√)
	G3: Automation and Tool support	×	(√)

TABLE I
ASSESSMENT CRITERIA AND EVALUATION RESULTS

of fulfillment by ITIL in contrast to NGOSS. A check mark means the aspect is fully or almost fully covered by the respective framework. A cross mark means the criterion is not sufficiently addressed or fulfilled. A check mark in brackets states that the aspect is addressed and partially fulfilled although essential sub-aspects are missing.

B. Results

The assessment matrix gives a good survey on where ITIL and NGOSS might complement one another, in which areas they overlap and what problems are not yet addressed by any of them. This section gives a summary and additional explanations.

1) Where are ITIL and NGOSS complementary?: Generally spoken, the strengths of ITIL lie in the management aspects. ITIL defines a clear SLM process as well as interfaces to other ITSM processes. Further on, ITIL defines a number of KPIs for evaluating the performance of the process itself. Since the NGOSS Handbook is not process-oriented, there is no overlap or contradiction in this field. Thus, from a management process perspective, ITIL and NGOSS can be used together. The NGOSS guidelines can be assigned to the activities provided by the ITIL SLM process. For example, the NGOSS SLA content recommendations can be applied within the third ITIL SLM process activity ("Contract").

2) Areas of overlap and potential conflicts: An overlap in the assessment table does not necessarily mean ITIL and NGOSS to be contradictory in the respective aspect. Examination of S3 and S4, the criteria with the most overlap, has proved that – again – ITIL is much more abstract than NGOSS in its recommendations and practices. Thus, in the end it can be said that there arise *no critical conflicts* when putting ITIL and NGOSS together. This makes it very attractive to use both ITIL and NGOSS as complementary frameworks in SLM.

3) Unaddressed challenges: For future work in the area of business-driven SLM, the areas of S1 (Mapping Support for QoS parameters), S5 (Support for SLA-based charging and accounting) and G3 (Automation and Tool support) have turned out as most challenging and at the same time most necessary in order to add the missing features to an environment of coexistence of ITIL and NGOSS.

In the area of QoS mapping (S1), various efforts have been undertaken in the past, but the fundamental problem of vertically mapping resource QoS to service QoS has not been solved in generality, since each partial solution of this problem needs to address the semantic specifics of a respective service or scenario. A promising approach is the development of a Service MIB that aims at making IT services manageable by adopting a management concept known from traditional Network and Systems Management – the Management Information Base (MIB) – to them [12].

With respect to Automation and Tool support (G3) there are several pieces of work some of which we shortly present in the next section. All have in common that they support specific elements in SLM and do not cover the entire process. For SLA-based accounting (S5) there are no feasible solutions available today.

VI. RELATED WORK

Besides the approaches presented in this paper, a lot of papers and white papers on the issue of SLM have been released in the last years. We selected ITIL SLM and the NGOSS Handbook because of their increasing popularity, while – at the same time – their roots are very different: The ITIL guidelines have been released from the British government as "Best Practices". By contrast, the NGOSS Handbook is a development of the Tele Management Forum which is composed of telecommunication enterprises, but also researchers in the area of IT and Telecommunication Service Management.

To our knowledge, business-aligned frameworks with a comparable scope as the one of ITIL and NGOSS do not exist. Surprisingly, neither ITIL nor NGOSS have proved as business-aligned in the way we described it in the criterion M4: considering the real monetary business impact of ITSM- or SLM-related decisions. In the year 2004, an interesting approach, covering particularly this field, has been published by HP and is called *Management by Contract (MbC)* [13]. Although MbC would not fit into our frameworks comparison, since it does not share the specifics a framework should entail, we give a very short survey on the objectives of MbC, because

it could be a suitable concept for filling this specific gap.

Management by Contract: This approach has been developed and wants to be understood as a paradigm for business-aligned IT Management. Its most important goal is to rationally meet and justify IT-related management decisions on the basis of contractual relationships, considering the business environment and impacts of IT management actions. Insofar, contracts and SLAs are not regarded as a *product of*, but more as a *basis for* IT Service Management.

Within the MbC architecture, SLAs play a quite decisive role, though their establishment does not matter for the approach. In the MbC architecture description, SLAs are characterized by the following three aspects:

- SLAs represent the requirements under which the service provider must deliver.
- The guarantees are negotiated prior to service deployment, but can be renegotiated over time.
- SLAs contain parameters of the service (availability and service latency are exemplary mentioned) as well as associated penalties and rewards for both parties.

This characterization is very close to the one we gave in Section II. The initial perspective of MbC is the so called conventional 3-layer IT Management Stack which consists of a Monitoring Layer, a Diagnosis Layer and a Recovery Planning Layer. MbC extends this model and adds a fourth layer: the Contract-based Analysis Layer. This one is meant to give a business context to the recovery options coming from the Recovery Planning Layer, reflecting the impact a recovery option would entail – based on the commitments specified in the SLAs.

Due to space restrictions, we refer to [13] for a summary of the MbC approach. This paper does not only describe the architecture in more detail, but also addresses the process of contract-based analysis and decision making taking place on the Contract-based Analysis Layer.

VII. Summary & Conclusion

In this paper, we analyzed ITIL Service Level Management and the NGOSS SLA Management Handbook as two important frameworks for business-driven SLM. By putting both approaches into a common context of service provisioning and applying a consistent terminology to them, we were able to show how they correlate to each other.

The ITIL and NGOSS approaches have shown as quite complementary which already provides an excellent starting point for further integration efforts. While ITIL proposes structure and content of the entire SLM process, NGOSS may be used in order to enrich this framework by valuable recommendations in specific partial aspects – especially the concrete design of SLAs.

The results of the comparison make one thing visible: Neither ITIL, nor NGOSS should compulsory be considered to be implemented exclusively in one environment. The analysis shows that there are many fields in which ITIL and NGOSS SLM are complementary to each other. Insofar, the relationship between ITIL and NGOSS is not "XOR". However, neither ITIL nor NGOSS gives sufficient guidance in the fields of SLA-based charging and accounting and Automation and Tool Support. An application note as it is available for eTOM and ITIL is not available for the NGOSS Handbook and ITIL SLM, but could be a helpful support for implementers and IT managers who want to adopt concepts from both ITIL and NGOSS.

Acknowledgment

The author wishes to thank the members of the Munich Network Management (MNM) Team for helpful discussions and valuable comments on previous versions of the paper. The MNM Team directed by Prof. Dr. Heinz-Gerd Hegering is a group of researchers of the University of Munich, the Munich University of Technology, the University of the Federal Armed Forces Munich, and the Leibniz Supercomputing Center of the Bavarian Academy of Sciences. Its web-server is located at http://www.mnm-team.org. This paper was supported by the EC IST-EMANICS Network of Excellence.

References

[1] OGC, Ed., *Introduction to ITIL*, ser. IT Infrastructure Library. The Stationary Office, 2005.

[2] Tele Management Forum (TMF), Ed., *SLA Management Handbook: Volume 1 - Executive Overview*, ser. NGOSS SLA Management Handbook. Morristown, USA: Tele Management Forum (TMF), 2005.

[3] Office of Government Commerce (OGC), Ed., *Best Practice for Service Delivery*, ser. IT Infrastructure Library (ITIL). Norwich, UK: The Stationary Office, 2001.

[4] B. Stiller and D. Hausheer, "State-of-the-Art in Economic Management of Internet Services," 2006.

[5] M. Garschhammer, R. Hauck, B. Kempter, I. Radisic, H. Roelle, and H. Schmidt, "The MNM Service Model — Refined Views on Generic Service Management," *Journal of Communications and Networks*, vol. 3, no. 4, pp. 297–306, Dec. 2001. [Online]. Available: http://wwwmnmteam.informatik.uni-muenchen.de/_php-bin/pub/show_pub.php?key=ghkr01

[6] T. Kaiser, "Methodology for the Determination of the Availability of Distributed, Application-oriented Services," Munich, Germany, Apr. 1999.

[7] *ISO/IEC 20000-1:2005 – Information Technology - Service Management - Part 1: Specification*, ISO/IEC, Dec. 2005.

[8] Office of Government Commerce (OGC), Ed., *Best Practice for Service Support*, ser. IT Infrastructure Library (ITIL). Norwich, UK: The Stationary Office, 2000.

[9] Tele Management Forum (TMF), "NGOSS (New Generation Operations Systems and Software) initiative," Object Management Group, Tech. Rep. telecom/01-04-13, Apr. 2001. [Online]. Available: ftp://ftp.omg.org/pub/docs/telecom/01-04-13.pdf

[10] Tele Management Forum (TMF), Ed., *SLA Management Handbook: Volume 2 - Concepts and Principles*, ser. NGOSS SLA Management Handbook. Morristown, USA: Tele Management Forum (TMF), 2005.

[11] ——, *SLA Management Handbook: Volume 3 - Service and Technology Examples*, ser. NGOSS SLA Management Handbook. Morristown, USA: Tele Management Forum (TMF), 2005.

[12] M. Sailer, "Towards a Service Management Information Base," in *IBM PhD Student Symposium at ICSOC05*, Amsterdam, Netherlands, Dec. 2005.

[13] C. Bartolini, A. Boulmakou, A. Christodoulou, A. Farrell, M. Salle, and D. Trastour, "Management by Contract: IT Management driven by Business Objectives," in *Proceedings of the 11th Workshop of the HP OpenView University Association (HPOVUA 2004)*, vol. 2004, Paris, France, Jun. 2004.

Admission Control for Inter-domain Real-Time Traffic Originating from Differentiated Services Stub Domains

Stylianos Georgoulas[1], George Pavlou[1], Panos Trimintzios[2], and Kin-Hon Ho[1]

[1] Centre for Communication Systems Research, University of Surrey
Guildford, Surrey, GU2 7XH, United Kingdom
[2] ENISA, EU, PO Box 1309, 71001, Heraklion, Crete, Greece

Abstract. Differentiated Services (DiffServ) are seen as the technology to support Quality of Service (QoS) in IP networks in a scalable manner by allowing traffic aggregation within the engineered traffic classes. In DiffServ domains, admission control additionally needs to be employed in order to control the amount of traffic into the engineered traffic classes so as to prevent overloads that can lead to QoS violations. In this paper we present an admission control scheme for inter-domain real-time traffic originating from DiffServ stub domains; that is real-time traffic originating from end-users connected to a DiffServ stub domain towards destinations outside the geographical scope of that domain. By means of simulations we show that our scheme performs well and that it compares favorably against other schemes found in the literature.

Keywords: Admission Control, Real-time Traffic, Differentiated Services.

1 Introduction

DiffServ offers a scalable approach towards QoS in the Internet by grouping traffic with similar QoS requirements into one of the engineered traffic classes and forwarding it in an aggregate fashion. To provide QoS guarantees, DiffServ domains must additionally deploy admission control in order to control the amount of traffic injected into the traffic classes so as to prevent overloads that can lead to QoS violations.

The various admission control schemes can be classified into three categories: endpoint admission control (EAC), traffic descriptor-based admission control (TDAC), and measurement-based admission control (MBAC). EAC is based on metrics applied to probing packets sent along the transmission path before the flow is established [1]. The probing packets can be sent either at the same priority as flow packets (in-band probing) or at a lower priority (out-of-band probing). One problem of EAC schemes is that simultaneous probing by many sources can lead to a situation known as thrashing [1]. That is, even though the number of admitted flows is small, the cumulative level of probing packets prevents further admissions. TDAC is based on the assumption that traffic descriptors are provided for each flow prior to its establishment. This approach achieves high utilization when the traffic descriptors used by the scheme are appropriate. Nevertheless, in practice, it suffers from several problems [2]. One is the inability to come up with appropriate traffic descriptors before establishing the flow. MBAC tries to avoid this problem by shifting the task of traffic characterization to the network [2].

F. Boavida et al. (Eds.): WWIC 2007, LNCS 4517, pp. 115 – 128, 2007.

That means that the network attempts to "learn" the characteristics of existing flows through real-time measurements. This approach has certain advantages. For example, a conservative specification does not result in overallocation of resources for the entire duration of the service session. Also, when traffic from different flows is multiplexed, the QoS experienced depends on their aggregate behavior, the statistics of which are easier to estimate. However, relying on measured quantities raises issues, such as estimation errors and memory related issues [2].

The various admission control schemes can also be classified according to the location where the admission control decision is made; at a centralized server or at various possible points in a network in a distributed manner. The idea of centralized schemes is simple. Signaling messages are exchanged between the sender of the flow and the centralized entity and between routers in the network and the centralized entity. These messages include the requirements of the flow and the resources state at each router, therefore admission control is performed by an entity that has complete and up-to-date knowledge of the network topology and resources, which is an ideal situation. However, in practice, centralized schemes have certain disadvantages. One is that a centralized entity constitutes a single point of failure. Another is the scalability problems that a centralized scheme raises [3]. Distributed schemes avoid these problems, but the existence of multiple admission control decision points means that concurrent admission control decisions may be made by distinct decision points for flows competing for the same resources; this can lead to QoS violations. In order for concurrency to be handled there exist some proposals in the literature [4], such as employing some safety margins to absorb the negative effects of concurrency.

Most of the schemes, to be applicable in practice, explicitly or implicitly make the assumption that the traffic is intra-domain; that is it originates and terminates within the same domain. The schemes that do not make this assumption, in many cases, e.g. see [5], require the cooperation of adjacent domains along the end-to-end paths on a per-flow basis as well as the existence of a commonly understandable signaling protocol end-to-end in order to perform admission control in each domain and propagate downstream the admission control decision and/or the QoS received so far.

Contrary to these schemes, in this paper we present a measurement-based admission control scheme for inter-domain real-time traffic originating from DiffServ stub domains, which when deployed in the context of a cascaded QoS peering model, does not require the cooperation and signaling among adjacent domains on a per-flow basis. In the rest we will first present the assumptions and conditions needed for this scheme to provide end-to-end QoS (section 2). We will then describe in detail our scheme (section 3) and we will evaluate and compare its performance against other schemes found in the literature (section 4) before concluding the paper in section 5.

2 Assumptions and Conditions

2.1 Existence of a Cascaded QoS Peering Model

The main assumption in our scheme is that a cascaded QoS peering model, similar to the one of the MESCAL project [6], is employed in the Internet. Each network provider or Autonomous System (AS) establishes provider service level agreements

(pSLAs) with the directly interconnected network providers. This type of peering agreement is used to provide QoS connectivity from a customer to reachable destinations several domains away. Fig. 1 gives an overview of the operations in this model.

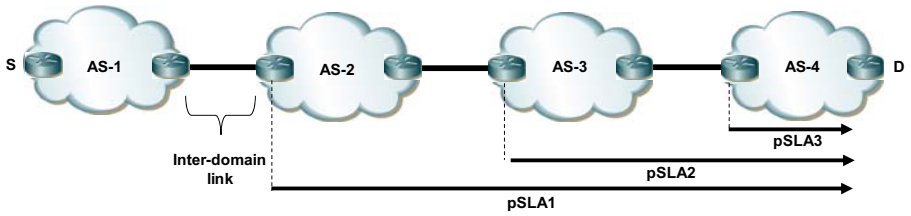


Fig. 1. A cascaded QoS peering model

AS-3 negotiates and establishes a peering agreement with AS-4 ($pSLA_3$) that will allow customers of AS-3 to reach destinations in AS-4 with specific QoS guarantees as long as the total aggregate demand from AS-3 does not exceed the negotiated and agreed bandwidth value in $pSLA_3$. AS-2, in turn, can negotiate with AS-3 a peering agreement ($pSLA_2$) in order to reach destinations in AS-4 with specific QoS guarantees. These guarantees are derived by combining the guarantees specified in $pSLA_3$ and the local QoS capabilities of AS-3. In a similar way, AS-1, which is the DiffServ stub domain, can establish a peering agreement $pSLA_1$ with AS-2 which defines the QoS guarantees that the traffic exiting AS-1 will receive from the ingress nodes of AS-2 till the end-customers connected to AS-4 as long as the aggregate demand from AS-1 does not exceed the negotiated and agreed in $pSLA_1$ bandwidth value.

Since pSLAs are established for aggregate demands, each network provider typically only has to manage a limited number for pSLAs, making the cascaded model scalable. By assuming that such a cascaded QoS peering model exists, DiffServ domain AS-1 does not need to cooperate and signal any of the downstream domains on a per-flow basis for traffic destined to remote destinations. It only needs to ensure that its inter-domain traffic does not exceed the negotiated bandwidth value in the corresponding pSLAs and that the QoS received by this traffic inside AS-1, when combined with the QoS values specified in the pSLAs is adequate to meet the end-to-end QoS requirements. The domains in a QoS chain just need to ensure they enforce the local QoS and that the traffic exiting them towards the next domain in this chain, is mapped to the appropriate class of the downstream domain according to the relevant pSLAs.

In this paper we focus on how the DiffServ stub domain AS-1 will ensure that the inter-domain real-time traffic originating from end-users of AS-1 will receive the required 'local' QoS treatment so that when combined with the QoS specified in the corresponding pSLAs it will still meet the end-to-end QoS requirements.

In the rest and for the sake of simplicity we will assume that towards the destinations of interest in a remote domain, AS-1 has one pSLA in place with AS-2 which specifies a bandwidth value C_{pSLA} and the associated packet loss rate PLR_{pSLA}, delay D_{pSLA} and jitter J_{pSLA} guarantees that will be met as long as the real-time traffic demand does not exceed the negotiated and agreed bandwidth value C_{pSLA}.

2.2 Local QoS Versus End-to-End QoS

Given that the delay and jitter parameters are additive and that, for low values, packet loss is also additive [5], and by knowing the end-to-end requirements regarding packet loss $PLR_{end-to-end}$, delay $D_{end-to-end}$ and jitter $J_{end-to-end}$ of the real-time traffic and also the relevant values agreed in the pSLA, it is straightforward to deduce the local QoS values that need to be enforced in the DiffServ domain AS-1. If we denote as PLR_{local} the local PLR requirement, as D_{local} the local delay requirement and as J_{local} the local jitter requirement, then these are given by:

$$\begin{aligned} PLR_{local} &\leq PLR_{end-to-end} - PLR_{pSLA} \\ D_{local} &\leq D_{end-to-end} - D_{pSLA} \\ J_{local} &\leq J_{end-to-end} - J_{pSLA} \end{aligned} \tag{1}$$

2.3 Enforcing Local QoS for Inter-domain Real-Time Traffic

We define as real-time traffic, sources that have strict *delay* and *jitter* requirements and a bounded *packet loss rate* (PLR) requirement. Regarding low delay and jitter, both requirements are likely to be met in a high-speed network core [7]. Furthermore, certain off-line traffic engineering actions can be taken so that delay and jitter are kept within low bounds. For example, the delay requirement can be taken into account by: a) configuring appropriately small queues for the real-time traffic in order to keep the per-hop delay small, and b) controlling routing to choose paths with a constrained number of hops. Jitter can remain controlled as long as the real-time traffic flows are shaped to their nominal peak rate at the network ingress [8]. Also, the deployment of non-work conserving scheduling can be beneficial for controlling jitter [9].

Given that a certain small amount of packet loss can be acceptable [7] without significant quality degradation and that delay and jitter can be controlled by taking the above actions, in this paper we employ the PLR as the QoS metric that needs to be controlled by the admission control scheme employed in the DiffServ stub domain and we focus on keeping it in values lower than the local PLR requirement.

2.4 Measurement/Enforcement Points

A measurement based admission control scheme needs to ensure that it controls the flow of traffic across all possible congestion points (bottlenecks).

As stated in [10], the edge links are currently considered as the most probable congestion points of a domain, whereas backbone links are overprovisioned. Therefore we assume that packets are lost at the DiffServ domain's ingress nodes, whereas in the core of the DiffServ domain, real-time traffic aggregates from different ingress nodes are treated in a peak rate manner. This means that the core is transparent to the real-time traffic sources with respect to packet loss. By assuming that the interior of the DiffServ domain has been engineered in this way and by taking into account the routing behavior, at each ingress node we can have an estimate of the bandwidth available for the inter-domain real-time traffic aggregate from that ingress (to be more precise, from that ingress node output interfaces) to each of the corresponding egress

nodes specified in the corresponding pSLA. For inter-domain traffic, however, one also needs to take into account that peering links at the border routers between domains are also bottlenecks [11], therefore they cannot be considered overprovisioned.

Taking the above into account, the proposed scheme applies actions at these bottleneck points (output interfaces of ingress nodes and output interfaces of egress nodes) and aims to ensure that the total PLR incurred at these points is less than the local PLR requirement for the inter-domain real-time traffic. This means that for each pair of ingress-egress nodes output interfaces the following condition is met:

$$\begin{aligned} &\forall(l(i),m(e)) \; with \; i \in I, e \in E, l(i) \in L_i, m(e) \in M_e \\ &and \; f(l(i),m(e)) = 1 : PLR_{l(i)} + PLR_{m(e)} \leq PLR_{local} \end{aligned} \tag{2}$$

where $l(i)$ is the output interface l of ingress node i, $m(e)$ is the output interface m of egress node e, I is the set of ingress nodes with end-customers generating real-time traffic towards the destinations in the pSLA, E is the set of egress nodes that are specified in the pSLA as exit points for inter-domain real-time traffic from the DiffServ domain towards the destinations in the pSLA, L_i is the set of output interfaces of ingress node i, M_e is the set of output interfaces of egress node e, $PLR_{l(i)}$ is the incurred PLR at the output interface l of ingress node i, $PLR_{m(e)}$ is the incurred PLR at the output interface m of egress node e and $f(l(i),m(e)) = 1$ indicates that the output interface l of ingress node i uses the output interface m of egress node e as the exit point towards the destinations in the pSLA.

We assume that as a result of the provisioning phase, these sets of ingress-egress pairs, as well as the output interfaces pairing and the bandwidth allocated within the domain are already known. We will also denote as $C_{l(i)\rightarrow m(e)}$ the available bandwidth for the inter-domain real-time traffic from the output interface l of ingress node i until the output interface m of egress node e, as $C_{m(e),pSLA}$ the available bandwidth from the output interface m of egress node e till the destinations specified in the pSLA, and finally as $C_{e,pSLA}$ the available bandwidth for the inter-domain real-time traffic from egress node e till the destinations specified in the pSLA in place.

We assume that it holds:

$$C_{m(e),pSLA} = UF_{m(e)} \times \sum_{i \in I : f(l(i),m(e))=1} C_{l(i)\rightarrow m(e)}, \; \forall e \in E \; and \; UF_{m(e)} < 1 \tag{3}$$

where $UF_{m(e)}$ is the underprovisioning factor, which indicates the extent to which the inter-domain links are underprovisioned with respect to the aggregate bandwidth reservations at the output interfaces of the ingress nodes. $UF_{m(e)}$ needs to be a number with value less than 1, otherwise the inter-domain links would not be bottlenecks.

We also assume that it holds:

$$\sum_{m \in M_e} C_{m(e),pSLA} = C_{e,pSLA} \quad and \quad \sum_{e \in E} C_{e,pSLA=}C_{pSLA} \tag{4}$$

In the next section we will present our scheme, which is distributed and does not require any cooperation between ingress nodes or any per ingress-egress operations or monitoring. It requires per-flow signaling only from the end-users till the ingress node of the DiffServ stub domain they are connected to but not further downstream and it tries to ensure that the local PLR requirement is met by regulating the admission of new flows but not by penalizing or terminating prematurely existing flows.

3 Admission Control Scheme

As stated in [12], in order for an admission control scheme to be successful in practice, it has to fulfill the following requirements.

- *Robustness:* A scheme must ensure that the requested QoS is provided. This is not trivial; for measurement-based schemes, measurement inevitably has some uncertainty, potentially leading to admission errors. The QoS should also be robust to traffic heterogeneity, long-range dependency, and to heavy offered loads.
- *Resource utilization:* The secondary goal for admission control is to maximize resource utilization, subject to the QoS constraints for the admitted flows.
- *Implementation:* The cost of deploying a scheme must be smaller than its benefits. In addition, the traffic characteristics required by the scheme should be easily obtained and the scheme should scale well with the number of flows.

3.1 Admission Control Logic

Our scheme consists of two modules, one module running at each ingress node i serving inter-domain real-time traffic from that node till each one of the egress nodes and one module running at each egress node e. The modules running at the ingress nodes make admission control decisions independently from each other, aiming to regulate the admission of new flows, based on feedback from the egress nodes modules.

The egress nodes modules continuously monitor the state of the egress output interfaces (to be more precise, the status of each of the output queues configured with bandwidth limit $C_{m(e),pSLA}$) and based on their status, at intervals of duration S they communicate PLR information to the ingress nodes that use these egress output interfaces as exit points for their inter-domain real-time traffic. This PLR information is used by the ingress nodes modules to calculate new PLR values to be used -if needed- for the admission of new flows. This means that each egress node only communicates with the ingress nodes that actually use them as exit points for their inter-domain real-time traffic. We need to clarify here that the communicated information relates to the PLR of the aggregate traffic using this egress output interface and not to the PLR of traffic originating from distinct ingresses, therefore the egress nodes do not need to keep any per-ingress state or perform any ingress-specific operations.

3.2 The Ingress Node Module

The functionality of the ingress node module is very similar to the functionality of the module in case of intra-domain traffic described in detail in [13].

We assume that every time an inter-domain real-time flow wants to be established, it signals this to the ingress node i. Then the module, based on a target PLR level $PLR_{l(i),target}$, decides to accept the flow establishment if the bandwidth $C_{l(i)\rightarrow m(e)}$ from that output interface l of ingress node i till the egress node e is enough in order to accommodate the existing flows and the new flow requesting admission, while at the same time satisfying this $PLR_{l(i),target}$ value. Since, as stated above, each egress node does not keep any per-ingress state and only communicates one PLR information per egress output interface, all $PLR_{l(i),target}$ values for all ingress nodes that use the same output interface of egress node e as exit point, should be the same. For the rest we will denote the PLR target at interface l of ingress node i, associated with the interface m of egress node e, as $PLR^{m(e)}_{l(i),target}$. This target $PLR^{m(e)}_{l(i),target}$ level is not fixed but is adjusted based on the feedback. Also, in order for the scheme to be able to recover the total locally incurred PLR to values less than the local PLR requirement without having to penalize or terminate existing flows, this $PLR^{m(e)}_{l(i),target}$ should be less than the local PLR requirement, that is:

$$PLR^{m(e)}_{l(i),target} \leq PLR_{local} \times OMF^{m(e)}_{l(i)} \ with \ OMF^{m(e)}_{l(i)} \in (0,1) \tag{5}$$

where $OMF^{m(e)}_{l(i)}$ is an Operational Margin Factor, defining the operational area within which $PLR^{m(e)}_{l(i),target}$ can range. $OMF^{m(e)}_{l(i)}$ should not be given a value close to one and the reason for this is that if, for example, $PLR^{m(e)}_{l(i),target}$ is allowed to get close or become equal to PLR_{local} and an overload situation occurs at the egress node output interface with bandwidth limit $C_{m(e),pSLA}$, then it may not be possible to recover the total locally incurred PLR to values less than the local PLR requirement just by regulating the admission of new flows, because the overload is caused by the existing flows and it will persist until some of the existing flows are terminated. In a similar manner, $OMF^{m(e)}_{l(i)}$ should not be set to very low values, because then the range [0, $PLR_{local} \times OMF^{m(e)}_{l(i)}$] within which $PLR^{m(e)}_{l(i),target}$ can range will be very limited, which will reduce the ability of the ingress nodes modules to react and regulate the admission of new flows, regardless of the feedback information.

3.3 The Egress Node Module

The egress node module passively monitors the output interfaces with bandwidth limit $C_{m(e),pSLA}$ (for the sake of simplicity, we will focus on one egress output interface and refer to it simply as output queue) and every S seconds (we will refer to S as the reporting period) it calculates the packet loss during the past interval of T seconds and depending on its value it reports back to the ingress nodes, which then adjust the target $PLR^{m(e)}_{l(i),target}$ level accordingly.

3.3.1 Egress Node Module Functionality

The desired functionality for the egress node module is to be able to react not abruptly but smoothly (still in a timely fashion) and provide feedback to the ingress nodes modules to regulate the admission of new flows. In order to achieve this smooth but timely operation, the egress node module when first senses a possible congestion situation, it initially tries to correct it by applying a set of 'mild' actions and if this situation is not resolved then it adopts more drastic 'emergency' measures.

In order to achieve this progressive operation, we define two threshold PLR values, named *soft threshold* and *hard threshold* respectively, against which $PLR_{m(e),T}$ is compared and depending on whether it crosses them (upwards or downwards) a specific set of actions is taken. The former threshold is denoted as soft, because it is allowed to be crossed upwards and still the status of the inter-domain link can be considered as not imminently close to becoming congested, whereas the latter is denoted as *hard*, because when it is crossed upwards, it means that the inter-domain link is imminently close to becoming congested. Since by employing the Operational Margin Factor, we have defined an upper value for the PLR allowed at the ingress nodes, both these thresholds should belong to the range $[0, PLR_{local} - PLR_{local} \times OMF_{l(i)}^{m(e)}]$.

3.3.2 Soft and Hard Threshold

The *soft threshold* $PLR_{m(e)}^{soft}$ is a PLR value, which, as long as it is not crossed upwards by $PLR_{m(e),T}$, no action is taken by the ingress node modules and no communication packets are sent. The range $[0, PLR_{m(e)}^{soft}]$ for $PLR_{m(e),T}$, therefore, corresponds to a 'normal operations' range. While in this range, the ingress nodes modules perform admission control using $PLR_{local} \times OMF_{l(i)}^{m(e)}$ as the $PLR_{l(i),target}^{m(e)}$ level.

The *hard threshold* $PLR_{m(e)}^{hard}$ is a PLR value that defines a range $(PLR_{m(e)}^{soft}, PLR_{m(e)}^{hard}]$, which indicates that a potential congestion situation may arise. While the measured $PLR_{m(e),T}$ is in this range, the egress node sends back to the ingress nodes communication packets that contain as information the difference between $PLR_{m(e),T}$ and $PLR_{m(e)}^{soft}$; that is the $PLR_{m(e),T} - PLR_{m(e)}^{soft}$ value. The ingress nodes receiving this value react to the potential congestion situation by adjusting the $PLR_{l(i),target}^{m(e)}$ level. In order for the ingress node modules to perform more conservative admission control as $PLR_{m(e),T}$ increases, we set the $PLR_{l(i),target}^{m(e)}$ level to be:

$$PLR_{l(i),target}^{m(e)} = PLR_{local} \times OMF_{l(i)}^{m(e)} - (PLR_{m(e),T} - PLR_{m(e)}^{soft}) \quad (6)$$

That is, the more the measured $PLR_{m(e),T}$ deviates from the soft threshold and approaches the hard threshold, the more conservative the admission control becomes. In practice, the ingress node modules attempt to compensate for these deviations by

decreasing by the same amount the $PLR^{m(e)}_{l(i),target}$ value. If, however, despite the regulation of the admission of new flows, $PLR_{m(e),T}$ continues to increase and crosses upwards the hard threshold, then the ingress node modules completely block all incoming admission requests until $PLR_{m(e),T}$ returns to a value lower than the hard threshold. If $PLR_{m(e),T}$ keeps decreasing and becomes lower than the soft threshold, then the $PLR^{m(e)}_{l(i),target}$ level is set equal to $PLR_{local} \times OMF^{m(e)}_{l(i)}$ and the egress node stops sending communication packets till the soft threshold is crossed upwards again.

This approach minimizes the control overhead, since communication packets are only sent when needed. However, if these packets cannot be guaranteed loss-free delivery, then the ingress node modules may erroneously translate the non-delivery of a communication packet as a recovery to the 'normal operations' range. In such cases, one alternative would be to have communication packets sent continuously every S seconds so that the ingress nodes can detect the loss of a packet.

3.4 On the Selection of the Parameter Values

3.4.1 The Reporting Period *S*

The reporting period S defines how up-to-date with the current egress node output queue status, the ingress node modules are.

The lower the value of S the more up-to-date the information the ingress node modules use when making admission control decisions. However, the lower the value of S, the higher the control overhead. Furthermore, a very low value of S will not allow the traffic contribution of the recently admitted flows to be depicted properly in the measured $PLR_{m(e),T}$ and, therefore, in the reported $PLR_{m(e),T} - PLR^{soft}_{m(e)}$ value.

On the other hand, when an ingress node module performs admission control for flows arriving within two reporting periods, it is not aware of the actual effect that each of these flows will have at the egress nodes output interfaces. Therefore, within a longer S seconds period, the higher the number of arriving flows requesting admission and as a consequence the higher the possibility of making erroneous admission control decisions. In order for this phenomenon to be minimized, egress routers should explicitly perform admission control on a per-flow basis.

Moreover, since the ingress node modules do not cooperate with each other, they may make concurrent admission control decisions. This means that every ingress node is not aware of the traffic contribution from the other ingress nodes towards the same egress node output interface during an S seconds period. And the longer this S seconds period, the higher the number of arriving flows, and, therefore, the higher the possibility for each ingress node to make erroneous admission control decisions. In order for concurrency to be accounted for in our scheme, where competition between ingress nodes takes place only for resources on the inter-domain links, we employ some safety margins when setting the soft and hard threshold values.

As a result of the above discussion we conclude that the value for reporting period S should be a compromise between the above mentioned contradicting requirements.

3.4.2 The Measurement Window *T*

A small value of T will have as an effect the egress node modules to react abruptly to bursts. Moreover, for low values of PLR, a small value of T will mean that the measured $PLR_{m(e),T}$ may not be representative of the real output queue congestion status. On the other hand, a high value of T will reduce the ability of the scheme to react to non-stationarities and will also introduce correlation between successive admission control decisions [2]. Therefore, the value of T should be a compromise between these contradicting requirements.

3.4.3 The Soft and Hard Thresholds

The soft and hard threshold values define three operation ranges, which are:

- [0, $PLR_{m(e)}^{soft}$], normal operation
- ($PLR_{m(e)}^{soft}$, $PLR_{m(e)}^{hard}$], potential congestion
- ($PLR_{m(e)}^{hard}$, $PLR_{local} - PLR_{local} \times OMF_{l(i)}^{m(e)}$], immediate congestion

Therefore, the value $PLR_{m(e)}^{soft}$ determines when the scheme will start reacting to increases in the measured $PLR_{m(e),T}$. The value $PLR_{m(e)}^{hard}$ determines when the scheme will start taking 'emergency actions' to heal immediately impending congestion situations and the difference $PLR_{m(e)}^{hard}$ - $PLR_{m(e)}^{soft}$ determines for how long the scheme will try to recover the system by applying 'mild' actions.

The $PLR_{m(e)}^{soft}$ value setting should take into account the $PLR_{local} \times OMF_{l(i)}^{m(e)}$ value, e.g. to guarantee that eq. 6 does not become negative before $PLR_{m(e),T}$ reaches the $PLR_{m(e)}^{hard}$ value. Also, the $PLR_{m(e)}^{hard}$ value, even though it could go up to $PLR_{local} - PLR_{local} \times OMF_{l(i)}^{m(e)}$, it should be set to lower values than that so as:

- To compensate for the effect of measurement errors.
- To compensate for concurrency-related issues.
- To allow the ingress node modules to react fast enough so that the local PLR requirement is met without having to penalize or terminate existing flows.
- To compensate for the fact that the exact effect of newly admitted flows on the status of the egress node output interfaces cannot be known beforehand. This is especially true, since the egress node modules are not aware of the traffic characteristics of individual flows.

To compensate for all the above, the practical solution we adopt is to set $PLR_{m(e)}^{soft}$ to a relatively low value and leave a margin between $PLR_{m(e)}^{hard}$ and $PLR_{local} - PLR_{local} \times OMF_{l(i)}^{m(e)}$.

4 Performance Evaluation

In order to evaluate the performance of our admission control scheme, we run simulations using the network simulator *ns-2* [14], with the topology of Fig. 2.

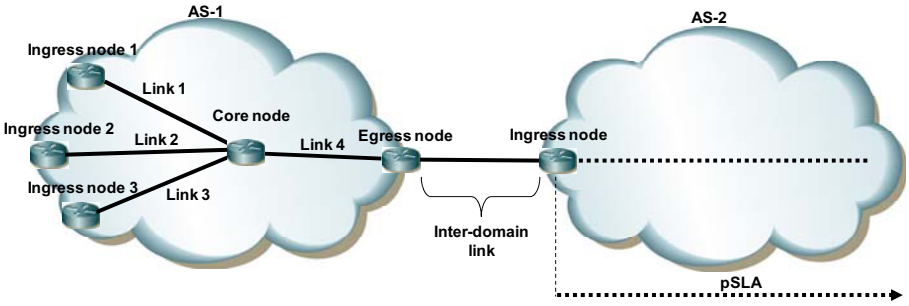


Fig. 2. Simulation topology

We use scenarios with the target local bound on PLR (PLR_{local}) for the inter-domain real-time traffic equal to 0.001. Since the value 0.01 defines a typically acceptable upper value of PLR for the VoIP service and for real-time applications in general [15], this implicitly means that the pSLA has to provide low, but not zero loss guarantees, to keep the end-to-end PLR below 0.01. We set the Operational Margin Factor for the ingress links 1-3 equal to 0.5, which means that the upper value that the target PLR at the ingress nodes output interfaces is allowed to get is equal to the half of the target local PLR, that is 0.0005.

We set the capacities allocated at links 1-3 for the inter-domain real-time traffic ($C_{l(i)\rightarrow m(e)}$) equal to 3.56Mbps. Since we assume that real-time traffic aggregates from different ingress node output interfaces are treated in the core in a peak rate manner, this means that the capacity allocated for the inter-domain real time traffic at link 4 is 10.68Mbps. We assume that the underprovisioning factor ($UF_{m(e)}$) is equal to 0.8, which means that the capacity allocated at the inter-domain link is 8.544Mbps. We also configure the queues at all links for the aggregate inter-domain real-time traffic to hold a maximum of 500bytes and we set the propagation delays at all links to be 5msec. For the sake of simplicity, we do not simulate the communication traffic, we do, however, consider the propagation delays from the instant it is generated at the egress node till the moment it can be used for admission control at the ingress nodes.

Regarding the algorithm's parameters, the employed values are: S = 1sec, T = 3sec, and we set the soft and hard thresholds equal to 40% and 60% of the $PLR_{local} - PLR_{local} \times OMF_{l(i)}^{m(e)}$ margin, which means that the employed value for the *soft* and *hard* threshold pair is (0.0002, 0.0003), meaning that 40% of the range $[0, PLR_{local} - PLR_{local} \times OMF_{l(i)}^{m(e)}]$ is left as safety margin.

In order to test the *robustness* of the scheme with respect to traffic *heterogeneity* and *long-range dependency*, we use a scenario with mixed VoIP and Videoconference

traffic sources, the same as in [13]. In order to test the *robustness* of the scheme with respect to *offered load*, as in [13] we test varying loading conditions ranging from 0.5 to 5, where the value 1 (*reference load)* corresponds to the average load that would be incurred by a source activation rate equal to 1000 VoIP sources/hour.

In order to compare the performance of our scheme, which we call inter-MBAC, against other schemes, we implement the EAC scheme described by Karlsson et al in [16]. Since this scheme (we call it EAC-KAR) is an out-of-band probing scheme, we implement a lower priority queue for the probing packets that can store, as in [17], a single probe packet. As in [16], we set the probing rate equal to the peak rate of the source requesting admission and we consider probe durations of 0.5sec up to 5sec. Since the path that needs to be probed includes the inter-domain link, we assume that the probing takes place between the ingress nodes 1-3 of AS-1 and the ingress node of AS-2, which after the end of the probing process signals back to the ingress nodes of AS-1 the PLR that the probing packets experienced. We do not simulate these signaling flows, we do, though, for fairness reasons consider the propagation delays.

As stated in [17], any admission control scheme must address the trade-off between *packet loss* and *utilization.* Therefore, for performance evaluation we use as metrics the locally incurred PLR and the utilization of the inter-domain link, which is the main bottleneck, together with the average blocking rate. For most loading conditions, EAC-KAR is not able to keep the total locally incurred PLR below the 0.001 local PLR target. The results shown are for 5 seconds of probe duration, which gives the lower violation of the local target PLR.

4.1 Simulation Results

Inter-MBAC satisfies the target local PLR for all loading conditions. We observe an increase in the incurred PLR for higher loading conditions, which is anticipated because it relies on measurements, so every new admission request has the potential of being a wrong decision [2]. Furthermore, this is due to concurrency related issues; the higher the load, the more flows arrive within every reporting period S.

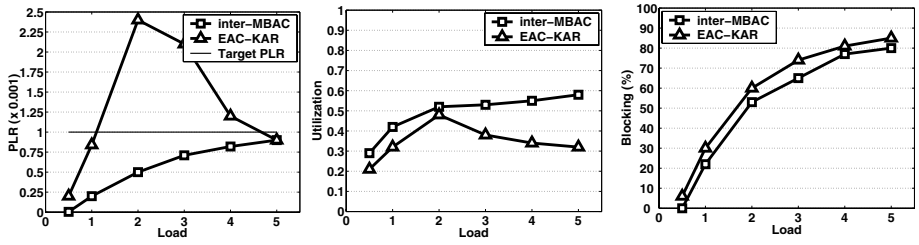


Fig. 3. Incurred PLR (left), inter-domain link utilization (centre) and blocking rate (right)

EAC-KAR violates the target local PLR for loading conditions more than one time the *reference load*. The trend of the incurred PLR for EAC-KAR indicates that it enters very early the thrashing region (for load more than two times the *reference load*) and despite the much higher (compared to inter-MBAC) incurred PLR, the achieved utilization is much lower and the incurred blocking is also higher. This behavior

seems to be a consequence of concurrency related issues which exaggerate the thrashing effect and create an oscillation effect. Flows are initially admitted, then because of the amount of probing packets, subsequent flows are rejected, the real-time traffic class is emptied, then a batch of flows is erroneously admitted (which justifies the violations of PLR), then the subsequent flows are rejected (which justifies the high blocking and the low utilization) and so on.

4.2 Further Discussion of the Simulation Results

The simulation results show that inter-MBAC can satisfy the target PLR for all tested loading conditions without requiring reconfiguration of its parameters for individual loading conditions. EAC-KAR fails to satisfy the local target PLR for most loading conditions despite reconfiguring its probe duration. The local target PLR is satisfied for very high load conditions but this is actually due to the thrashing effect.

Regarding the control overhead, it is not straightforward to compare the two schemes using an absolute metric since we have not implemented the communication process or the signaling control process for EAC-KAR. However, we can state that since for EAC-KAR the control overhead is dependent on the number of flows, whereas for inter-MBAC the control overhead is dependent not on the number of flows but on the number of edge nodes, the control overhead of inter-MBAC is expected to be less than that of EAC-KAR in real network situations.

Moreover, for our simulation setup, for inter-MBAC and for low loading conditions (less than load equal to the *reference load*) the simulations show that no communication packets need to be sent back to the ingress nodes because the soft threshold value is not violated at any time. Therefore, there is no control overhead associated with inter-MBAC at very low loading conditions. For higher loading conditions, the control overhead increases and for loading conditions more than one time the *reference load* it stabilizes, since its frequency is determined by the reporting period S and not by the flow arrival dynamics. For EAC-KAR, there is control overhead at all loading conditions and it increases proportionally with the load.

5 Conclusions

In this paper we presented a measurement based admission control for inter-domain real-time traffic originating from DiffServ stub domains.

We showed through simulations that the scheme is *robust* to traffic heterogeneity, time-scale fluctuations and heavy offered loads. The scheme can meet the QoS objectives for a variety of loading conditions without requiring any reconfiguration of its parameters and without incurring significant control overhead. Furthermore, the scheme achieves satisfactory *utilization* and compares well against existing admission control approaches for the same simulation setup.

Our scheme is also easy to *implement*. It is distributed and does not require any cooperation between ingress nodes. Per-flow operations are only performed at the ingress nodes, and egress nodes do not need to keep any per-flow state or perform any per-flow or ingress-specific operations. The scheme requires per-flow signaling only from the end-users till the ingress node of the DiffServ stub domain they are

connected to. Also since it is makes the assumption that a hop-by-hop cascaded QoS peering model between adjacent domains exists, it does not require any cooperation of adjacent domains along the end-to-end paths on a per-flow basis or the existence of a commonly understandable signaling protocol end-to-end.

Acknowledgments. This work was undertaken in the context of the IST ENTHRONE phase 2 and IST EMANICS projects, which are partially funded by the Commission of the European Union.

References

1. L. Breslau et al. "Endpoint Admission Control: Architectural Issues and Performance", SIGCOMM 2000.
2. M. Grossglauser et al. "A Framework for Robust Measurement-Based Admission Control", IEEE/ACM Transactions on Networking, June 1999.
3. C. Chuah et al. "Resource Provisioning using a Clearing House Architecture", IEEE IWQoS 2000.
4. S. Lima et al. "Distributed Admission Control in Multiservice IP Networks: Concurrency issues", Journal of Communications, June 2006.
5. S. Lima et al. "Distributed Admission Control for QoS and SLS Management", Journal of Network and Systems Management, September 2004.
6. M. Howarth et al. " Provisioning for Interdomain Quality of Service: the MESCAL Approach", IEEE Communications Magazine, June 2005.
7. G. Schollmeier et al. "Providing Sustainable QoS in Next-Generation Networks", IEEE Communications Magazine, June 2004.
8. T. Bonald et al. "Statistical Performance Guarantees for Streaming Flows using Expedited Forwarding", IEEE INFOCOM 2001.
9. M. Mowbray et al. "Capacity Reservation for Multimedia Traffics", Distr. Syst. Eng., 1998.
10. V. Padmanabhan et al. "Server-based inference of Internet Link Lossiness", IEEE INFOCOM 2003.
11. T. Bressoud et al. "Optimal Configuration for BGP Route Selection", IEEE INFOCOM 2003.
12. M. Grossglauser et al. "A Time-Scale Decomposition Approach to Measurement-Based Admission Control", IEEE/ACM Transactions on Networking, August 2003.
13. S. Georgoulas et al. "Heterogeneous Real-time Traffic Admission Control in Differentiated Services Domains", IEEE GLOBECOM 2005.
14. K. Fall et al. "The *ns* manual" (www.isi.edu/nsnam/ns/ns_doc.pdf).
15. T. Chaded, "IP QoS Parameters", TF-NGN, November 2000.
16. V. Elek et al. "Admission Control based on End-to End Measurements", IEEE INFOCOM 2000.
17. R. Gibbens et al. "Measurement-based connection admission control", 15th International Teletraffic Congress, June 1997.